Optimising Quality of Service Levels through Experimentation on Streaming Multimedia Applications using WiMAX.

by

Amanda Peart

The thesis is submitted in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy of the University of Portsmouth

School of Computing
University of Portsmouth
Lion Terrace, Portsmouth, Hampshire
PO1 3HE, United Kingdom

November 2014

Abstract

This work investigates the optimisation of Quality of Service (QoS) levels through experimentation on streaming multimedia applications with WiMAX. This work initially tests two algorithms, the intelligent Adaptive Buffer Control (iABC) and the Bandwidth Fluctuation Location Based Dynamic Transmission Rate Control before bring these together in the Resource Harvest and Redistribution - Call Admission Control (RHR-CAC) algorithm.

The iABC algorithm augments ATM QoS categories within the wireless domain. iABC utilizes an intelligent adaptive buffering system for the existing users to redistribute the resources to facilitate further connections. iABC results demonstrates that additional connections can be accommodated while maintaining the QoS needs for the existing connections, improving the efficient use of resources. Whereas, the bandwidth fluctuation location-based dynamic transmission rate-limit control framework predicts when a user enters a weaker signal area and dynamically limits the bandwidth of other users to facilitate the QoS needs when the signal strength is reduced. The framework made its predictions based on location and mobile network coverage map queries before initiating the resource rate-limiting algorithm. This significantly improved QoS in video streaming and smoothing of bandwidth fluctuation. These two algorithms have initiated success with bandwidth management of harvesting bandwidth for one user without affecting the QoS of the others.

It is important to determine when simulating any QoS protocol, whether the perceived improvement will actually function under the planned usage, consequently it is vital to replicate the reality of the users behaviour. It is not just the technological issues that affect connectivity but also physical mobility. Therefore the geographical perspective of the user’s physical location impacts on the relevance of experiments, to ensure they reflect reality as closely as possible. Therefore mobility were investigated to select the best fit for this work. The results illustrate that the random waypoint or random walk models both emulated the reality of a user in a mobile environment, for a simulation.

The work culminates, with the RHR – CAC algorithm which builds on the previous algorithms within the WiMAX domain. This encompasses all QoS, service flows by harvesting resources from those provisioned but not activated to provide the resources to accept additional connections. This improves the efficient redistribution of resources while improving the connectivity rate at the call admission control. The results evidenced that QoS has been maintained for all current connection while ensuring that all service flow are not starved of bandwidth.

Table of Contents

		Declaration	**13**
		Nomenclature	**14**
		Acknowledgements	**19**
		Dissemination	**20**
Chapter	**1**	**Introduction**	**22**
1.1		The social Technical need for communication	**22**
1.2		The Internet: An Ever Evolving Global Communication System	**24**
1.3		Technological Overview of the Connection Medium	**26**
	1.3.1	IEEE 820.11	**28**
	1.3.2	WiMAX IEEE 802.16	**29**
	1.3.3	Next Generation ‘G’ Telecommunication	**30**
	1.3.4	3G	**30**
	1.3.5	4G	**32**
1.4		Defining Quality of Service	**33**
1.5		Motivation	**35**
1.6		Problem Statement	**36**
1.7		Contribution	**36**
	1.7.1	iABC – Adaptive Buffer Control	**36**
	1.7.2	Mobile Network Bandwidth Fluctuation Location-Based, Dymanic Transmission Rate=Limit Control	**37**
	1.7.3	Real World Aspects of Social & Envirnomental Influences on User Mobility Within the WiMAX Domain	**38**

	1.7.4	Real Time Bandwidth Redistribution Transmission Algorithm	**39**
	1.7.5	Resource Harvest Redistribution Call AdmissionControl Algorithm (RHR-CAC)	**40**
1.8		Research Methodology	**41**
	1.8.1	The Scientific Method	**42**
	1.8.2	Computer Simulation	**43**
	1.8.3	Interleaving of the Hypothetical-Deductive Model and the Simulation Study Methodology	**46**
1.9		Summary	**46**
Chapter	**2**	**Quality of Service within the Wireless Domain**	**52**
2.1		Introduction to Quality of Service	**53**
2.2		Network Technologies that support QoS	**55**
2.3		IEEE 802.11: The Wireless QoS Evolution	**58**
2.4		IEEE 802.16 Standard: The Quality of Service Perspective	**64**
2.5		Evolution of the IEEE 802.16 Standard	**66**
2.6		Quality of Service Mechanisms built into the IEEE 802.16	**67**
2.7		PHY Layer: Modulation	**69**
	2.7.1	Adaptive Modulation	**70**
	2.7.2	Adaptive Bursts Profiles	**71**
2.8		IEEE 802.16: QoS Provisioning	**72**
2.9		Service Flow Categorizations	**73**
	2.9.1	Unsolicited Grant Service (UGS)	**75**
	2.9.2	Real-time Polling Service (rtPS)	**75**
	2.9.3	Extended real-time Polling Service (ertPS)	**76**

2.10	Dynamic Service Establishment	**77**
2.10.1	Two-Phase Activation Model	**78**
2.10.2	QoS Parameter Sets	**78**
2.10.3	Authorization Modulation	**79**
2.11	IEEE 802.16 MAC Layer	**80**
2.11.1	Convergence SubLayer	**81**
2.12	Scheduling	**82**
2.12.1	Bandwidth Requests	**82**
2.12.2	Bandwidth Grants	**83**
2.13	Scheduling Algorithms	**85**
2.13.1	Balanced Fairness and Throughput Algorithms	**88**
2.13.1.1	Round Robin	**88**
2.13.1.2	Maximum Signal Noise Ratio	**89**
2.13.1.3	Proportional Fair Algorithms	**90**
2.13.1.4	Adaptive Proportional FairnessAlgorithm	**91**
2.13.2	Weighted/Deficit Calculations Based Scheduling Algorithm	**92**
2.13.2.1	Generalised Processor Shaping	**93**
2.13.2.2	Tuncated Generalised Processor Shaping	**94**
2.13.2.3	Weighted Round Robin	**95**
2.13.2.4	Weighted Fair Queuing	**96**
2.13.2.3	Deficit Round Robin	**97**
2.13.2.4	Weighted Round Robin	**97**
2.13.2.7	Earliest Deadline First	**97**

	2.13.2.8	Highest Urgency First	99
	2.13.3	Opportunistic and Cross-layer Algorithm	100
	2.13.3.1	Carrier to Noise Ratio	101
	2.13.3.2	Maximum Carrier to Noise Ratio	103
	2.13.3.3	Normalised Carrier to Noise Scheduling	103
	2.13.3.4	Opportuntistic Round Robin	105
	2.13.3.4.1	Cross Layer Algorithm	105
	2.13.3.4.2	Modified Largest Weighted Delay First (MLWDF)	105
6	2.13.4	Hierachical and Hybrid	106
2.14		Summary	111
Chapter	**3**	**Call Admission Control (CAC) Schemes**	**113**
3.1		Call Admission Provisioning	115
3.2		Call Admissions Control: The Constraints	119
3.3		Summary	128
Chapter	**4**	**Pilot Scenarios: Resource Redistribution**	**129**
4.1		Pilot Study 1: The viability of Redistributing Bandwidth Access to Improve connection Rates in WATM	129
	4.1.1	ATM Networks	130
	4.1.2	Wireless ATM (WATM)	131
	4.1.3	Enhancing QoS for Wireless	131
	4.1.3.1	Mobiware	131
	4.1.4	Intelligent Adaptive Buffer Control an Extension to Mobiware	134
	4.1.4.1	Intelligent Adaptive Buffer Control	134

4.1.5		Scenario: Bandwidth Distribution	**138**
4.1.6		Summary: Pilot Study 1: Bandwidth Distribution	**138**
4.2		Pilot Study 2: Predicting Mobile Neywork Bandwidth Fluctuation to Enhance Video Stream Service Quality	**139**
4.2.1		Maintain QoS when moving from a SSA to a WSA region	**142**
4.2.2		Performance Testing	**147**
4.2.2.1		Test Scenario 1	**147**
4.2.2.2		Test Scenario 2	**147**
4.2.2.3		Test Scenario 3	**151**
4.2.3		Performance of Framework (Scenario 3)	**153**
4.2.4		Test Analysis	**154**
4.3		Summary	**154**
Chapter	**5**	**Resource Harvest and Redistribution CAC Algorithm: Improving the Ratio of Successful Connections**	**155**
5.1		Resource Harvest and Redistribution CAC Algorithm	**156**
5.2		Bandwidth Redistribution UGS Simulation	**163**
5.3		Bandwidth Allocation and Utilization Simulation	**165**
5.4		User Mobility in Adhoc Networks	**169**
5.5		Random Walk	**170**
5.6		Random Waypoint	**174**
5.7		Guass-Markov	**176**
5.8		Environmental Effects on Mobility	**178**
5.9		Social Theory Effects on Mobility	**180**
5.10		Environmental Model	**181**

5.11	RHR-CAC Analysis	**182**
5.12	Summary	**186**
Chapter 6	**Conclusion**	**190**
6.1	Limitations of the Research	**192**
6.2	Future Work	**192**
	References	**199**
	Appendices	**209**
	Ethics UPR16	**209**
	Random Walk Code Snippetts	**211**
	Random Walk with IPv4	**213**
	Random Walk 2 Nodes	**222**
	Random Walk 4 Nodes	**226**
	Random Walk 8 Nodes	**231**
	Random Walk 32 Nodes	**241**

List of Tables

Table 1-1 Internet and Worldstats : Usage and Population Statistics **23**
Table 1-2 Internet and Worldstats : Usage and Population Statistics **23**
Table 1-3 Comparison of the Theoretical and real world implementations of current and future network technologies **31**
Table 2.1 Comparison of Best Effort, DiffServ and IntServ service Models **54**
Table 2.2 The Evolution of the IEEE 802.11 standard **56**
Table 2.3 Evolution of the IEEE 802.16 Standard **65**
Table 2.4 Modulation and Coding Explained **67**
Table 2.5 Summary of types of modulation with data capacities **69**
Table 2.6 Modulation and FEC Profiling **70**
Table 2.7 Bandwidth request mechanisms per services flow type **81**
Table 4.1 Variable Values for Test Scenario 1 **145**
Table 4.2 Data for b_k best-case users illustrated in Figure 5.7 **147**
Table 4.3 Results of algorithm (Figure 4.9) on $B_k$users (Table 4.2.) **148**
Table 4.4 T_2, B_n Test Values for Test Scenario 2 **149**
Table 4.5 Data for B_n Best-Case Users **149**
Table 4.6 T_3, B_z Details for Test Scenario 3 **151**
Table 5.1 Parameters used in the simulations for the Random Walk mobility model **169**
Table 5.2 An Example of an Adjacency Relationship Matrix between 5 nodes **170**
Table 5.3 The Application and SF Type for Connections **181**

List of Figures

Figure 1.1 Technology Comparisons **25**
Figure 1.2 Network Evolutions **25**
Figure 1.3 Line of Sight Verus Non-Light of Sight **28**
Figure 1.4 3G and 4G frequency Spectrums and data Capacity **29**
Figure 1.5 The iterative nature of the Hypothetico-Deductive Method **41**
Figure 1.6 Steps in a discrete simulation study **44**
Figure 1.7 Mind map of the key stages of the research being Undertaken **45**
Figure 1.8 Depicts a conceptual model of how the key features of the simulation study model, Hypothetico-Deductive Method and the mind map **47**
Figure 2.1 Latency: The New Web Performance Bottleneck **52**
Figure 2.2 ATM Protocol Reference model **55**
Figure 2.3 Point Coordination Function and Distributed Coordination Function Architecture **58**
Figure 2.4 Hidden Station Problem **59**
Figure 2.5 Mapping in EDCF in IEEE 802.11e **60**
Figure 2.6 An Example of phase delay that if excessive will render an audio application useable **63**
Figure 2.7 Comparison of 16-QAM, 64-QAM and 256 QAM **68**
Figure 2.8 WiMAX modulation and coding schemes illustrating throughput and distance parameters. **69**
Figure 2.9 Architecture of the Subscriber Station Illustrating the Scheduling Services **71**
Figure 2.10 Dynamic Service Allocation (DSA) **75**
Figure 2.11 The relationship of QoSParamSet of resource allocation per variable **77**
Figure 2.12 IEEE 802.16 Implementation of the Reference Model with WiMAX **79**
Figure 2.13 IEEE 802.16 QoS Architecture **84**
Figure 2.14 Taxonomy of Scheduling Algorithms for WiMAX
Figure 2.15 Round Robin scheduling example **85**
Figure 3.1 IEEE 802.16 QoS Architecture **112**
Figure 3.2 Hierarchical structure of bandwidth allocation **115**
Figure 3.3 Overview of establishing resource reservation for a BS to SS connection **118**

Figure 3.4 Example of BS to SS Communication **119**
Figure 3.5 Call Admissions Control Algorithm **123**
Figure 4.1. Mobiware Architecture **130**
Figure 4.2. An Example of iABC **133**
Figure 4.3 Effects of Transmission Periods on Delay **136**
Figure 4.4 Mobile video stream pixilation due to being starved of data **138**
Figure 4.5 Client-side algorithm (pseudocode) to predict user entering a WSA **138**
Figure 4.6 Client-side algorithm (Flow-Chart) to predict user entering a WSA **140**
Figure 4.7 Server-Side algorithm to determine which users to rate (pseudocode) **141**
Figure. 4.8 Server-Side algorithm to determine which users to rate (Flow-Chart) **143**
Figure 4.9 Mathematical formula to determine suitable maxRLC for Figure 4.5 **144**
Figure 4.10 Client-side algorithm to best utilize a granted bandwidth boost **145**
Figure 4.11 Demo area with T_1 (purple and B_k nearest best-case users *(blue))* **146**
Figure 4.12 Performance of proposed dynamic rate-limit **151**
Figure 5.1 Resource Harvest and Redistribution – CAC Algorithm **159**
Figure 5.2 Bandwidth Utilisation **164**
Figure 5.3 Allocated Bandwidth **164**
Figure 5.4 Route of a Typical Mobile Device User **169**
Figure 5.5 Random Walk Mobility Model Simulating Two Nodes **170**
Figure 5.6 Random Walk Mobility Model Simulating Sixteen Nodes **171**
Figure 5.7 Random Walk Mobility Model Simulating Thirty Two Nodes **172**
Figure 5.8 Typical Movement Produce for the Random Waypoint…… **173**
Figure 5.9 Single Node Mobility using the Random Waypoint Model. **174**
Figure 5.10 A One Node Trace of a Mobile Node using the Gauss-Markov Mobility Model **175**
Figure 5.11 The possible angle increments for movement **177**
Figure 5.12 Implementation of the proposed model when a node (N) is faced with an obstacle impeding its desired path to its goal (G) **177**
Figure 5.13 The average node densities of RWP and the proposed model with a varying number of obstacles. **179**

Figure 5.14 The node density values for RWP and the proposed model with 5 obstacles **180**
Figure 5.15 Maximum Instantaneous gross data rate **180**
Figure 5.16 Maximum payload gross data rates **181**
Figure 5.17 FIFO and WFQ scheduling comparision for granted connections **183**
Figure 5.18 FIFO scheduling after RHR-CAC **188**

Declaration

Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.

SIGNED:__ _ _______

Nomenclature

A number of key terms are used throughout this document and are defined here:

3G	Third Generation
4G	Forth Generation
AIFS	Artibitration Interframe Space
AP	Access Point
AQR	Atomatic Repeat Request
ARPANET	Advanced Research Projects Agency Network
ATM	Asynchronous Transfer Mode
BE	Best Effort
BER	Bit Error Rate
bps	bits per second
BS	Base Station
BU	Bandwidth Utilization
CAC	Call Admissions Control
CBP	Connection Blocking Probability
CBR	Constant Bit Rate
CCK	Complementary Code Keying
CDP	Connection Dropping Probability
CF	Contention Free
CFP	Contention-Free Period
CID	Connection ID
CNR	Channel to Noise Ratio
CNR i.i.d	Carrier to noise ratio independent and identically distributed
CP	Contention-Period
CSMA/CD	Carrier Sense Multiple Access With Collision Detection
Cspec	Channel Specification
CTS	Clear To Send
CW	Contention Window
CWmax	Contention Window Maximum
CWmin	Contention Window minimum
DCF	Distributed Coordination Function
DiffServ	Differentiated Services

DL	Down Link
DOCSIS	Data Over Cable Service Interface Specifications
DRC	Data Rate Control
DRR	Deficit Round Robin
DSA	Dynamic Service Activate
DSA-REQ	Dynamic Service Activate-Request
DSA-RSP	Dynamic Service Activate-Response
DSC	Dynamic Service Change
DSD	Dynamic Service Delete
DSSS	Direct sequence spread spectrum
EDCA	Enhanced Distributed Channel Access
EDCF	Enhance Distributed Coordination Function
EDF	Earliest Deadline First
EM	Envirnomental Model
FCFS	First Come First Service
FDD	Frequency Division Duplex
FEC	Forward Error Correction
FFT	Fast Fourier Transforms
FHSS	Frequency-hopping spread spectrum
FIFO	First In First Out
FTP	File Transfer Protocol
GOS	Grade of Service
GPC	Grant per Connection
GPS	Generalized Processor Shaping
GPSS	Grant per Subscriber Station
HARQ	Hybrid Automatic Repeat-reQuest
HCF	Hybrid Coordination Function
HoQ	Head of Queue
HUF	Highest Urgency First
iABC	Intelligent Adaptive Buffer Control
IEEE	Institute of Electrical and Electronics Engineers
IntServ	Intergrated Services
IP	Internet Protocol
IP-TV	Internet Protocol television
IPv4	IP version 4
IPv6	IP version 6
ITU-T	International Telecommunication Union-Telecommunication

MAC	Media Access Control
Max CNR	Maximum Channel to noise Ratio
MaxSNR	Maximum Signal to Noise Ratio
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
MLWDF	Modified Largest Weighted Delay First
MNCM	Mobile Network Coverage Map
MPLS	Multiprotocol Label Switching
MRR	Minimum Reserved Requirements
MRTR	Minimum Reserved Traffic Rate
MSTR	Maximum Sustained Traffic Rate
NCR	Normalised Channel to noise ratio
NLOS	Non Line of Sight
nrtPS	non real-time Polling Service
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiplexing Access
ORR	Opportuntistic Round Robin
PCF	Point Coordination Function
PDU	Protocol Data Unit
PGPS	Packet Generalized Processor Shaping
PIF	PCF Interframe Function
PLT	Page Load Times
PMP	Point-to-Multipoint
PSTN	Public Switched Telephone Network
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
QTBR	Quadra Threshold Bandwidth Reservation
RHR-CAC	Resource Harvest Redistribution Call Admission Control
RR	Round Robin
rtPS	real-time Polling Service
RTS	Request To send
RW	Random Walk
SDU	Service Data Unit
SFID	Service Flow ID
SINR	Signal to Interference-plus-Noise Ratio
SNR	Signal to Noise Ratio
SS	Subscriber Station

SSA	Strong Signal Area
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TGPS	Truncated Generalized Processor Sharing
Tspec	Traffic specification
TT	Transportation Theory
TXOP	Transmission Opportunity
TXOP	Transmission Opportunities
UDP	User Datagram Protocol
UGS	Unsolicited Grant Services
UL	Up Link
UL-MAP	Uplink Map Message
VBRVS	Variable Bit Rate Video
VLAN	Virtual LAN
VoIP	Voice over IP
WDRR	Weighted Deficit Round Robin
WFQ	Waited Fair Queuing
WiMAX	Worldwide Interporability fo Mobile Internet Access
WLAN	Wireless Local Area Network
WMANS	Wireless Metropolitan Area Networks
WRR	Weighted Round Robin
WSA	Weak Signal Area

To my much-loved family Alan, Vikki and Danny

Acknowledgements

Studying for a PhD part-time means over the years many people have contributed indirectly to this thesis via their support, through general discussions; ideas, friendships, and many words of encouragement. There are many more people who have contributed than I could hope to name, to you all I appreicate the help you have all provided.

First and foremost I would like to thank my family for their continuing support in everything I do, without them I would not be where I am today. Despite the pressures and the commitment required to undertake this work they have always been there for me never wavering in their support and understanding in my drive to succeed. To my two children Vikki and Danny who have foregone valueable family time when I have needed to study, I am enternally grateful for your patience and support which have never faultered. To my husband Alan who is always supportive in what ever I choose to do, I thank you sincerely.

Also big thank you, to Dr Alice Good, Mrs Jacqui Allen-West and Dr Tineke Fitch for their encouragement that goes above and beyond what most friends would do to support me during my research, especially Jacqui's never ending supply of tea., Also my expression of appreciation towards my supervision team, Dr. Mo Adda and Dr Carl Adams, for their enthusiasm and guidance throughout my PhD

Also I would like to thank my colleagues Thanos, Gareth, Mohammad, and Adamtini for their support during the early days of the network research group. Lastly, I would like to thank all my friends for their encouragement and the enjoyment they brought to my life during all these years. Mel, Jayne, Tanya, Louise and all those whom I have missed out, thank you for every single second of your time.

Dissemination

Peart, Amanda, Lockett, A. and Adda, Mo (2013) *Predicting mobile network bandwidth fluctuation to enhance video stream service quality implementation of location-based, dynamic transmission rate-limit control.* In: IEEE Technically Co-Sponsored Science and Information Conference 2013, 2013-10-07 - 2013-10-09, London.

Peart, Amanda and Adda, Mo (2013) *Quality of service in WiMAX: real world aspects of social & environmental influences on mobility.* In: 9th Annual International Conference on Information Technology & Computer Science, 2013-05-20 - 2013-05-24, Athens.

Peart, Amanda and Good, Alice (2012) *Wireless bandwidth management authentication improving quality of service.* In: International Scientific Conference Computer and Solutions in Manufacturing Engineering – CoSME, 2012-10-01, Brasov.

Peart, Amanda, Adda, Mo and Goodman, A. (2012) *A QoS real time bandwidth redistribution transmission algorithm in WiMAX.* In: Papadopoulos, Y. and Petratos, P., eds. Enterprise management inforrmation systems :. Atiner, Ather Greece, pp. 63-71. ISBN 9789609549608

Peart, Amanda and Good, Alice (2012) *Wireless bandwidth authentication improving quality of service.* Academic Journal of Manufacturing Engineering, 10 (3). pp. 1-11. ISSN 1583-7904

Peart, Amanda, Adda, Mo and Goodman, A. (2011) *A QoS real time bandwidth redistribution transmission algorithm in WiMAX.* In: 7th Annual International Conference on Computer Science and Information Systems, 2011-06-13 - 2011-06-16, Athens.

Peart, Amanda and Adda, Mo (2009) *Quality of service: dynamic authentication bandwidth management for the wireless environment.* ICISE '09 Proceedings of the 2009 First IEEE International Conference on Information Science and Engineering. pp. 5366-5369. 10.1109/ICISE.2009.862

Peart, Amanda and Adda, Mo (2008) *Improved quality of service utilising high priority traffic in HCF in a dynamically changing wireless networks.* In: Petratos, P. and Dandapani, P., eds. Current advances in computing, engineering and information :. Atiner, Athens, Greece, pp. 195-209. ISBN 9789606672347

Peart, Amanda and Adda, Mo (2008) *Quality of Service challenges for Voice over Internet Protocol (VoIP) within the wireless environment.* In: Petratos, P. and Dandapani, P., eds. Recent Advances in Computing and Management Information Systems :. Atiner, Athens, Greece, pp. 221-234. ISBN 9789606672347

Adda, Mo, Peart, Amanda and Wakins, N. (2006) *Quality of service in wireless ATM for high demand multimedia applications.* In: Information and Communication Technologies, 2006. ICTTA '06. 2nd. Vol. 2 :. IEEE, New York, pp. 3233-3238. ISBN 0780395212

Adda, Mo, Peart, Amanda and Wakins, N. (2006) *Quality of Service on wireless ATM.* In: 6th International Network Conference, 2006-07-01, Plymouth.

Adda, Mo, Owen, Gareth, Kasassbeh, M. and Peart, Amanda (2005) *Communication issues in large scale wireless ad hoc networks.* In: Petratos, P. and Michalopoulos, D., eds. Computer Science and Information System :. Atiner, Athens, Greece, pp. 299-313. ISBN 9608867231

Chapter 1 Introduction

This chapter introduces the overarching themes of this thesis and places the motivation for the work into context. Thereafter, the rationale and goals are defined for the main QoS investigation of the project which are discussed, followed by a summary of the overall project.

1.1 The Social Technical need for Communication

The human need to communicate is intense we function as social beings; communication has become instinctive and is essential to build an informed society. Hailey, (2009) states '*Human communication is a multifaceted collection of systems that follow a snowball effect. Language allows us to communicate faster and simpler, but the language also elicits more language, thoughts, and ideas, which guides humans to devising improved methods (or worse ways) to carry out our communications through our social interactions*.' (Hailey, 2009). Montagu (1979) believes that human communication covers a multitude of signs that is more than media and messages, information and persuasion; it also meets a deeper need and serves a higher purpose (Montagu, 1979). He continues to state that communication is the grounds of meeting the foundation of community. '*It is, in short the essential human connection*' (Montagu, 1979). Therefore finding better and more efficient ways to communicate feeds this basic human need and in turn creates knowledge.

The mode of communication has taken various forms which have evolved over many years, from the pre-electric period of cave paintings providing a linear form of relaying messages through to the technological times of today's Internet. Communication has been a vital tool in the human's evolutionary development. We have an innate need to share information to gain knowledge and in turn build communities that is not constrained by physical space. '*Communication is so important that it is deemed necessary for physical health. In fact, evidence suggests that an absence of satisfying communications can even jeopardize life itself … personal communication is essential for our well-being.*' (Adler & Rodman, 2006). Dunbar (1996) suggests the first function of human communication is gossip (Dunbar, 1996), this has evolved from person to person, to being emulated on a larger scale via social networking sites such as Facebook© and Twitter© where the gossip is now far reaching and instant. Communities can take many forms from a small collection of like-minded people to those of national or international dimensions. This has been possible due to the Internet enabling instant communication that is no longer restricted to the geographical or physical ability to reach a particular area. Communities can communicate virtually without the constraints of physical locations.

The global business community has driven the need for more remote access to a communication infrastructure (Philpot, Beaton, & Whiteduck, 2014). This need aided the growth of the mobile workforce and a constant obligation to be contactable. Initially the mobile workforce would input data, e.g. orders etc. into their device and download this information periodically (daily or even weekly) from a wired infrastructure. The advent of the wireless domain gave business an even greater capability to improve communication with their mobile workforce, providing push email communication and instant access to business data to aid business decision making. This phenomena has also spilled into social aspects of life and has become the core medium for social communication (Borge-Holthoefer, Baños, & Moreno, 2013).

The ability to communicate globally has aided many innovations and collaborations worldwide. The backbone to this success is an adequate communication infrastructure, and to be fully efficient the mode of communication has to be '*interoperable, fast and delivered with integrity*' (Cabinet Office, UK, 2013). The demand for a perpetually connected digital information society drives the need for continued improvements for Quality of Service (QoS) within the mobile communication environment. The evolution of the wireless broadband domain has increased bandwidth availability, but 21st century multimedia applications are bandwidth hungry with an ever increasing demand for even more bandwidth. Therefore it is essential to develop more efficient algorithms to improve bandwidth utilisation together with the user experience.

1.2 The Internet: An Ever Evolving Global Communication System

The Internet has grown from being predominately an academic and military tool into an ever evolving global communication phenomenon with over 2.4 billion users worldwide. Table 1.1 illustrates the breakdown of internet users across the world in comparison to the world's regional population, (Internet World Stats, 2013) which has risen by 0.3 billion in the last year alone (Worldstats, 2011), see table 1.2. The demand for connectivity has seen both the wired and wireless environments experiencing rapid market adoption. The wireless environment alone has seen subscribers increase from 1 billion worldwide in 1990 to 2 billion in 2005 (ITU, 2005) and by 2013 this figure has risen to in excess of 8.8 billion active subscribers to both the wireless and mobile (cellular) broadband domain (ITU, 2013). The most significant increase in subscribers has been in developing countries such as Africa and the Middle East, which have seen an exponential increase in users. This enormous growth coupled with the requirements of multimedia applications is continuing to drive the demand for higher Internet speeds (Andrews, Ghosh, & Muhamed, 2007). The QoS connectivity experience of a wired Internet connection as a desktop user, has created demand for the same level of QoS to be emulated within the wireless domain. As, with all innovations, there are inherent problems to be resolved, especially when the underlined technology has to be deployed for mass market use e.g. having 'the *capability to provide resource assurance and service differentiation in a network*' (Wang, 2001).

WORLD INTERNET USAGE AND POPULATION STATISTICS June 30, 2012						
World Regions	Population (2012 Est.)	Internet Users Dec. 31, 2000	Internet Users Latest Data	Penetration (% Population)	Growth 2000-2012	Users % of Table
Africa	1,073,380,925	4,514,400	**167,335,676**	15.6 %	3,606.7 %	7.0 %
Asia	3,922,066,987	114,304,000	**1,076,681,059**	27.5 %	841.9 %	44.8 %
Europe	820,918,446	105,096,093	**518,512,109**	63.2 %	393.4 %	21.5 %
Middle East	223,608,203	3,284,800	**90,000,455**	40.2 %	2,639.9 %	3.7 %
North America	348,280,154	108,096,800	**273,785,413**	78.6 %	153.3 %	11.4 %
Latin America / Caribbean	593,688,638	18,068,919	**254,915,745**	42.9 %	1,310.8 %	10.6 %
Oceania / Australia	35,903,569	7,620,480	**24,287,919**	67.6 %	218.7 %	1.0 %
WORLD TOTAL	**7,017,846,922**	**360,985,492**	**2,405,518,376**	**34.3 %**	**566.4 %**	**100.0 %**

Table 1-1 Internet and Worldstats : Usage and Population Statistics (Internet World Stats, 2013) (Worldstats, 2013)

WORLD INTERNET USAGE AND POPULATION STATISTICS March 31, 2011						
World Regions	Population (2011 Est.)	Internet Users Dec. 31, 2000	Internet Users Latest Data	Penetration (% Population)	Growth 2000-2011	Users % of Table
Africa	1,037,524,058	4,514,400	**118,609,620**	11.4 %	2,527.4 %	5.7 %
Asia	3,879,740,877	114,304,000	**922,329,554**	23.8 %	706.9 %	44.0 %
Europe	816,426,346	105,096,093	**476,213,935**	58.3 %	353.1 %	22.7 %
Middle East	216,258,843	3,284,800	**68,553,666**	31.7 %	1,987.0 %	3.3 %
North America	347,394,870	108,096,800	**272,066,000**	78.3 %	151.7 %	13.0 %
Latin America / Carib.	597,283,165	18,068,919	**215,939,400**	36.2 %	1,037.4 %	10.3 %
Oceania / Australia	35,426,995	7,620,480	**21,293,830**	60.1 %	179.4 %	1.0 %
WORLD TOTAL	**6,930,055,154**	**360,985,492**	**2,095,006,005**	**30.2 %**	**480.4 %**	**100.0 %**

Table 1-2 Internet and Worldstats : Usage and Population Statistics (Worldstats, 2011).

The need from the perspective of the business sector is outlined via the Federation of Small Businesses who argue that "… *that the UK's broadband market needs to ensure that fit-for-purpose connectivity is available to everyone, regardless of location, and that it not only meets current demand but is also future-proofed. We define 'fit-for-purpose' broadband services as having speeds that are* *guaranteed as advertised, a high QoS, and symmetry between download speeds – which are prioritised by residential users – and upload speeds – which are equally if not more critical for businesses that operate online*" (Federation of Small Businesses, 2014)

1.3 Technological Overview of the Connection Medium

Developed countries such as the United Kingdom and United States have a good fixed wired broadband service, this has generally evolved from the telecommunication system. In these countries it is only areas that do not get an adequate fixed wired service, or the service is very expensive, that are more likely to source connectivity via a wireless broadband infrastructure such as rural Britain, though nine streets in Britain have recorded broadband at less than 1mbps (Allan, 2014). Whereas in countries such as China, India, Pakistan, Indonesia and Korea for example, who have not previously invested in developing a sophisticated fixed wired infrastructure or the nature of the geography of the country make deploying a fixed wired infrastructure prohibitive (Intel Information Technology-Gerard Smyth, 2006), the wireless domain is highly developed and has provided an efficient finanical route for their governments and industry's to catch up with the connected world.

Figure 1.1 illustrates that Wireless LANs (WLAN) supports a satisfactory level of data rates but has a limited range, which constrains mobility especially at vehicular speed. The mobile network provides greater coverage and mobility than the WLAN infrastructure. The drawback is the data rates delivered are low (but increasing with 4G) but the cost limits its accessibility to the general population, though the industrial competition to control the connectivity medium is driving down costs.

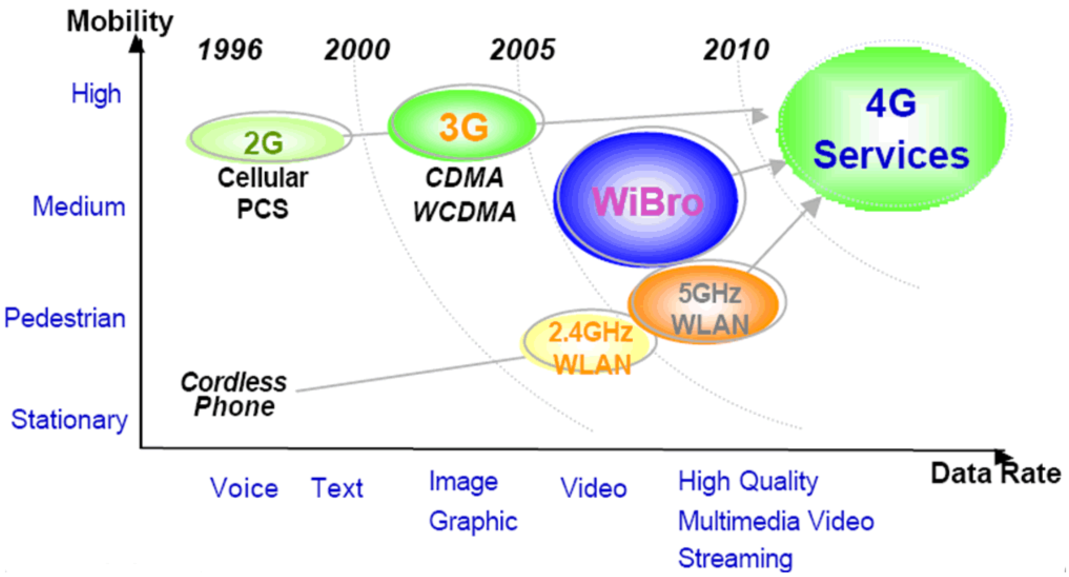

Figure 1.1 Network Evolutions (Soon Young Yoon, 2004)

WiBro (Korean version of IEEE 802.16) currently delivers the satisfactory balance between WLAN and Cellular networks of key network deliverables such as QoS, coverage (currently covering the 2.3GHz spectrum), data rates and mobility (vehicular speed 60km/h). It is envisaged that WiBro will evolve into the 4G network with convergence of voice, multimedia with mobility capabilities while still providing good QoS (see figure 1.2). Though with the emphasis on 4G deployment rather than WiMAX in the UK convergence has not yet been a realisation.

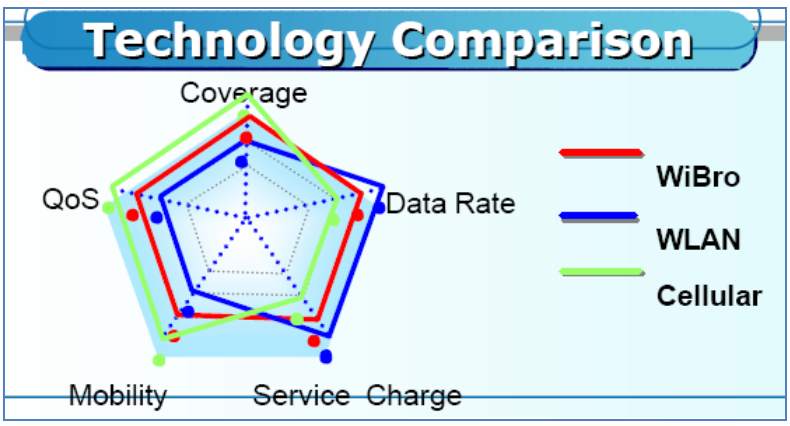

Figure 1.2 Technology Comparisons (Soon Young Yoon, 2004)

1.3.1 IEEE 802.11

In striving to meet the QoS demand and to aid interoperability the IEEE 802.11 working group was formed in 1990 (IEEE, nd), with the aim to develop a standard to aid all computers to communicate equally within the wireless domain. The first IEEE 802.11 standard was ratified in 1997 as the data communication format for WLAN. Wireless technologies have been improving to provide quality access for all kinds of applications. The driver has been from industry to develop solutions for the mobile workforce to maintain communication with their headquarters, which are largely used as access networks in an emerging markets (Gunasekaran & Marmantzis, 2005), (Lin, Ngo, & Qiao, 2006). Also eGovernment and the military need have influenced the evolution of the communication infrastructure. Vint Cerf states (Rosenfeld, 2000) '*People often take the view that standardization is the enemy of creativity. But I think that standards help make creativity possible -- by allowing for the establishment of an infrastructure, which then leads to enormous entrepreneurialism, creativity, and competitiveness*' (Rosenfeld, 2000). Standards facilitate the interoperability and openness for system connectivity for all.

The IEEE 802.11 standard strives to incorporate elements of QoS service and has resulted in multiple revisions. The IEEE 802.11a was the first iteration of the standard to incorporate QoS capabilities, for example traffic type prioritization, and contention-free transmission (IEEE, 1997). The drawback of the aim is that it is reliant on the Ethernet protocol and is not as interoperable as the 802.16 standard. The 802.11n introduced further QoS capabilities with priorities of different traffic types and contention-free transmission over short periods of time. But the limitation of Ethernet means that some of the QoS capabilities are not available (Sweeney, 2006). The MAC is contention based and therefore the acknowledgement mechanism which incorporates timeouts introduces high overheads and latency which reduces the QoS required by applications.

1.3.2 WiMAX IEEE 802.16

The Worldwide Interoperability for Microwave Access (WiMAX) forum was formed in June 2001 (WiMAX Forum, 2001-2014), which evolved the IEEE 802.16 Standard. Its ethos is to provide a wireless broadband service to the end user anywhere providing an improved QoS than currently available with Wi-Fi. The Fixed-WiMAX implementation was adapted from the Korean version of WiBro (wireless broadband), (Marks, 2006). The mobility was formally introduced to the specification in 2005 and later Multiple-Input Multiple-Output (MIMO) was incorporated into the specification in 2011, to further improve the QoS to the end user. WiMAX is set to challenge the boundaries of broadband access via the wireless infrastructure. Wi-Fi has the limitations of both speed and physical signal range, while functioning under the connectionless environment that provides users questionable levels of QoS. Whereas WiMAX operates at higher speeds with a greater physical coverage providing connection oriented connectivity to the end user, implying an improved QoS experience. It has the potential to improve the coverage of connectivity in current UK black spots, such as rural Britain. It is believed that WiMAX can rival cable broadband in both connectivity and QoS.

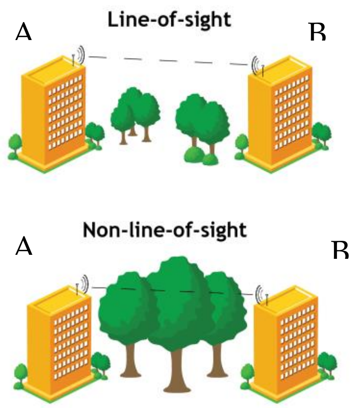

Figure1.3 Line of Sight versus Non-line of Sight (Avalan, 2014)

The non-line-of-sight (see figure 1.3) implementation of WiMAX provides a superior communication system to that provided by Wi-Fi especially in urban areas where it can accommodate diffraction of the signal around physical structures. Also the line-of-sight (see figure 1.3) implementation operates with a stronger signal providing more bandwidth and therefore better QoS. A typical Wi-Fi signal does not function at these frequencies (see section 2.4), nor can its signal compete with the physical coverage area of WiMAX (Wi-Fi 100m, WiMAX 50km). Together with the number of user connection that WiMAX can accommodate without degradation of the signal is more than can be accommodated within the Wi-Fi deployment.

1.3.3 Next Generation ‘G’ Telecommunications

The term “generation” or ‘G’ refers to a non-backwardly-compatible mobile telecommunication technology with the expectation of better performance and connectivity than the previous generation, benefits of the new infrastructure will only be gained if a compatible device is used (figure 1.4). This is not always the case with all internet compliant infrastructures, which provide interoperability (Beavis, 2013). Though this technology is not the direct focus of this work it is included for the overview of the mobile domain.

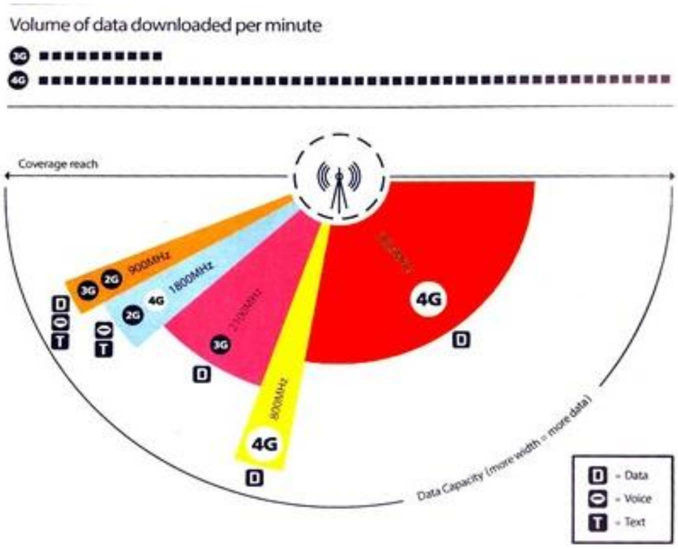

Figure 1.4 3G and 4G frequency Spectrums and data Capacity (Beavis, 2013)

1.3.4 3G

3G offers quality with audio with data rates of at least 200kbps raising to the throughput of broadband, though it is debated that it still does not challenge IP networks on the versatility of its usefulness, table 1.3. The ITU states "*it is expected that IMT-2000 will provide higher transmission rates: a minimum data rate of 2 Mbit/s for stationary or walking users, and 384 kbit/s in a moving vehicle,*" (ITU(a), 2005). Mapping QoS onto the circuit-switch layer can cause inefficiencies in delivery with scheduling and queuing issues. The adaptive modulation that Orthogonal Frequency Division Multiplexing (OFDM) offers is further advanced than High Speed Downlink Packet Access (HSDPA), though it can dynamically adapt to network conditions it is still a constraint. IEEE 802.16 can provide a comparable QoS if not better and is cheaper to deploy in adopted counties, mainly due to its connection oriented approach. To utilize 3G device specific technology is required this incurs additional cost to the user, whereas WiMAX can be accessed from any device providing the wireless card is compatible.

1.3.5 4G

The International Telecommunications Union-Radio communications sector (ITU-R) specified a set of requirements for 4G standards, to enable connectivity to ultra-broadband Internet. This incorporates peak speed requirements of 100 Mbps for high mobility vehicular communication, with a lower mobility boundary requirement of 1Gbps to be International Mobile Telecommunications Advanced (IMT-Advanced) 2008 compliant (ITU-R, 2013).

The 4G technology diversifies from the traditional IP telephony switched protocol that previous generation and relies on an IP protocol which utilizes the OFDMA transmission scheme to enable high bit rate transmission.

In the UK OFCOM auctioned off the frequency for 4G in three spectra, 800Mhz, 1800Mhz and 2600Mhz, table 1.3. The 800MHz allows longer range transmission that can carry everything from HD voice to high speed data More importantly it has the potential to resolve the dilemma of getting broadband to those rural communities currently without or experiencing limited connectivity (OfCom, 2013).

The 1800MHz spectrum is designed to offer a balance between coverage and speed. It provides significant improved performance while achieving good connectivity across the UK. While the 2600MHz is good for short ranges providing multiple simultaneous connections, this spectrum is best suited for crowded environments (Plummer, 2012).

		Real World (avg)		Theoretical (max)		Availability
		Download	Upload	Download	Upload	
2.5G	GPRS	32-48Kbps	15Kbps	114Kbps	20Kbps	Today
2.75G	EDGE	175Kbps	30Kbps	384Kbps	60Kbps	Today
3G	UMTS	226Kbps	30Kbps	384Kbps	64Kbps	Today
	W-CDMA	800Kbps	60Kbps	2Mbps	153Kbps	Today
	EV-DO Rev. A	1Mbps	500Kbps	3.1Mbps	1.8Mbps	Today
	HSPA 3.6	650Kbps	260Kbps	3.6Mbps	348Kbps	Today
	HSPA 7.2	1.4Mbps	700Kbps	7.2Mbps	2Mbps	Today
Pre-4G	WiMAX	3-6Mbps	1Mbps	100Mbps+	56Mbps	Today
	LTE	5-12Mbps	2-5Mbps	100Mbps+	50Mbps	End 2010
	HSPA+	-	-	56Mbps	22Mbps	2011
	HSPA 14	2Mbps	700Kbps	14Mbps	5.7Mbps	Today*
4G	WiMAX 2 (802.16m)	-	-	100Mbps mobile / 1Gbps fixed	60Mbps	2012
	LTE Advanced	-	-	100Mbps mobile / 1Gbps fixed	-	2012+

Table 1-3 Comparison of the Theoretical and real world implementations of current and future network technologies (Vilches, 2010).

1.4 Defining Quality of Service

Quality is synonymous with excellence, but is excellence actually what is required? Over engineering a system to achieve excellence can take time and incur excessive costs. Therefore the commercial perspective defines quality in terms of 'satisfying the user's needs' alternatively the military view is that the system should be 'fit for purpose'. Cisco networking (2003) defines QoS as the "Capability of a network to provide better service to selected network traffic over various technologies". Whereas the ITU-T defines QoS as "A set of quality requirements on the collective behaviour of one or more objects. Cisco and the ITU-T view QoS from different perspectives, Cisco's approach to QoS is determined by the physical network to provide a service, whereas the ITU-T defines quality as the requirements of the object, this could be the need of an application such as VoIP to function across a network satisfactorily –(Tung et al, 2008). Another view of QoS is the Grade of Service (GoS) which incorporates connection capacity, network coverage to include guaranteed maximum blocking probability and outages probability, this perspective is adopted from the telecommunications industry(ITU-T study group 2, 2007). IEEE 802.16 is designed to work with many data types e.g. streamed multimedia, VoIP as well as simple data transmission, with these applications users demand a satisfactory level of service to maintained the usability of the application.

1.5 Motivations

The human need for communication is now far reaching, today's requirement for 24/7 connectivity coupled with ever evolving technology where the technological boundaries are quickly reached. Along with the need to connect is the quality of that connection, coupled with the ever growing demands from multimedia applications that necessitate such quality to ensure their usability.

The continued evolution of the wireless infrastructure promises great improvements in the area of QoS. To date limited studies have focused on QoS priority differentiation though QoS provisioning has been investigated (Hong et al, 2006, Zhang et al, 2007b). Resource management, traffic shaping and access control is not currently in the definitive IEEE 802.16 standard (see figure 3.1). Though QoS signalling mechanisms are included, there are no algorithms defined on how to use the signals for bandwidth allocation and differentiated QoS (Ganz 2004). To maintain satisfactory levels of QoS guarantees for various applications while still achieving high systems utilization, the QoS architecture needs to be integrated into the MAC protocols (Chu 2002).

The current generation of users have discarded the wired infrastructure and now demand the freedom and mobility of connectivity that the wireless network environment provides. But along with this freedom, QoS issues that have long been resolved in the wired environment present themselves within the wireless domain. There is a fine line between providing satisfactory QoS to ensure that applications are usable or not. Streamed multimedia applications can suffer from jitter, pixilation and dropped frames, along with latency that cause lag, e.g. lip synchronisations problems. With VoIP a slight latency can cause an echo on the connection again making the application unusable.

Most research investigations have focused on either the scheduling or the Call Admissions Control (CAC) areas of QoS, therefore a gap in the research on combining these concepts and presenting an entire architecture is evident, as discussed in chapter 3. To-date research on QoS within the WiMax environment has been conducted with a focus either on scheduling or CAC mechanisms to improve QoS, either from the perspective of the BS or the SS.

Wongthavarawatt & Ganz (2003) conducted the first investigation into WiMAX comprising both scheduling and admissions control. They demonstrated the relationship between traffic characteristics, the required QoS requirements and traffic performance (Wongthavarawat & Ganz, 2003). Their view on the admissions control policy was to differentiate the scheduling algorithm used for each SF while maintaining current QoS for other connections utilizing real-time applications. To achieve this, the arrival time and packet deadline are determined to ascertain the maximum delay for the transmission. Wongthavarawatt & Ganz (2003) works ensures that packets meet their deadlines within the constraints of the finite bandwidth (Wongthavarawat & Ganz, 2003). For this algorithm results were evidenced only for rtPS and BE traffic with the assumption of all traffic being admitted. This work focused predominantly on the differentiation of the QoS traffic classes or Service Flow's (SF's) (see section 2.9). This thesis builds on the concept of differentiation of SF's with regards to the scheduling algorithm and also improves the utilization of the finite bandwidth, via redistribution of resources for all SF's.

The hypothesis of this work is '***Can WiMAX deliver the QoS guarantees that the end user demands in a mobile adhoc envirnoment***'. To achieve this, the principal contribution of this research is to define a framework to enhance the QoS for the IEEE 802.16 standard, utilizing the mechanisms and provisions that currently have been encompassed within the standard and improve their overall efficiency. This will incorporate improved scheduling from both the BS and the SS perspective, utilising the traffic classes to prioritise efficient admissions control mechanisms while ensuring efficient use of resources to provide increased connectivity without compromising quality.

1.6 Problem Statement

The aim of this research is to determine '*How current levels of Quality of Service can adequately be maintained for real-time streaming multimedia applications in an ad hoc wireless network environment*'.

1.7 Contribution

This section outlines the key area of contributions of this work, demonstrating an incremental approach to ascertaining the QoS within WiMAX.

1.7.1 iABC – Adaptive Buffer Control

This research focused on the evolution of the ATM paradigm facilitating wireless technology to encompass the advantages of the QoS provided by the

ATM priority algorithms that was previously absent within the wireless domain. The significant factor is that the ATM environment shares QoS similarities with WiMAX therefore the concepts can be modified for the WiMAX domain. Multimedia applications require a minimal level of service to function, and provide seamless transition between levels of service.

The intelligent Adaptive Buffer Control (iABC) system augments ATM's QoS properties within the wireless infrastructure (Adda, Peart, & Watkins, 2006). The system builds on the Mobiware system, which provides QoS support that allows multimedia applications to operate transparently during handoff and through heavy QoS requirement fluctuations. Mobiware is a middleware platform designed to run between the radio link layer and the application layer (Campell, 1998). The results demonstrate that iABC can maintain QoS within a saturated network, while permitting short-term use for additional applications via incorporating Mobiware's 'flow adaptive policy' with application intelligence. Mobiware guarantees the QoS though a secured bandwidth, the drawback being that this is a finite resource, iABC dynamically alters the size of the receiving applications buffer once additional demands arise. iABC provides an interesting proposition to provide all parties with their respective requests, while maintaining a high and maintainable QoS for all. Ultimately, bandwidth can only support a certain number of transmissions with an adequate level of quality. When that limit is reached, additional requests will simply have to wait for freed resources or be denied service. The ideology of this work is evident with the Resource Harvest and redistribution algorithm of this work (see chapter 6).

1.7.2 Mobile Network Bandwidth Fluctuation Location-Based, Dynamic Transmission Rate-Limit Control

Due to the very nature of modern day smartphones and tablets, users of such devices will often travel from an area with strong mobile signal to a weaker area. Travelling from a strong signal area (SSA) to a weak signal area (WSA) causes a significant drop in the mobile network bandwidth available to the device (Peart, Lockett, & Adda, 2013). A sudden bandwidth drop often causes the stream to rapidly become starved of buffered data, triggering a pause in playback resulting in QoS problems.

This work counters the mobility problem by attempting to predict a user entering a WSA, and dynamically rate-limiting other nearby best-case users to increase available bandwidth to the said user. Predictions are based on active user location information, and Mobile Network Coverage Map (MNCM) queries. The results demonstrated a significantly improved QoS of a video stream to a user entering a WSA. Though the framework is resource intensive it does demonstrate that improvements in smoothing mobile network bandwidth fluctuations can improve the QoS for users entering a WSA. This work contributes to the bandwidth management principles of constraining bandwidth for one user without affecting their QoS to maintain the QoS of another improves the overall efficiency of the system (chapter 6).

1.7.3 Real World Aspects of Social & Environmental Influences on User Mobility Within The WiMAX Domain (Peart & Adda, 2013)

In today's technological world, users of mobile wireless devices are predominantly on the move while still enjoying connectivity of the Internet. How people use their mobile devices differ in many ways, not only from a technological point of view, but also from a geographical point of view, the user's physical location. Theoretically modelled nodes have an uninterrupted straight path to their next destination in simulations, whereas in the real world this is extremely unlikely to be true with the average human meandering down the street, while concentrating on their mobile device. It is important to determine through simulating the proposed QoS protocols with WiMAX connectivity, whether the perceived improvement will actually function under the planned usage, consequently it is therefore vital to replicate the reality of user behaviour. This work investigates a variety of mobility models including Transportation Theory (TT), Random Walk (RW), Random Waypoint (RWP), and Gauss Markov models, as well as developing an Environment Model and how it affects connectivity within WiMAX (Peart, Lockett, & Adda, 2013). Camp et al (2002) recommend the use of either the RWP or RW mobility models if an entity mobility model is required. Key indicator to this decision is the availability of the models within the simulation packages and that with a few shortcomings mobility of the real-world to an expectable level is replicated (Camp, Boleng, & Davies, 2002). Results produced concurred with camp et al (2002), recommendation; both models can emulate movement which typifies mobile users in an ad hoc network.

The Environmental Model (EM) introduced social and environmental factors into the mobility model simulation. The constraint of this model is that any social connections together with the destination needs to be known before commencing the simulation which adds more computational requirements. In developing new algorithms the model will not add value to the initial simulation results, but if the system is deployed data can be collected and actual results can be calculated when bounded by a particular case study such as Dartmouth campus then this will add value (chapter 6).

1.7.4 Real Time Bandwidth Redistribution Transmission Algorithm

The 'bandwidth redistribution algorithm' was developed to build on the current WiMAX QoS class scheme aimed at improving bandwidth utilization for real time traffic with packets of a fixed size such as VoIP (Peart & Adda, 2011).

To improve QoS the bandwidth reallocation algorithm tracks the bandwidth usage for the transmission on a periodic basis and compares this to the total amount of bandwidth that the SS has allocated to that connection. If there is a significant difference, between the bandwidth allocated and the bandwidth being utilised for that connection (assuming that there are redundant pre-allocated resources) the bandwidth allocated can be reduced, allowing the redundant bandwidth to be redistributed to a new connection.

This enhancement worked on cutting down allocated bandwidth until a sufficient balance of resources was achieved. The bandwidth redistribution algorithm results evidenced that if a specified percentage of reserved bandwidth was redistributed (until the bandwidth was at least 80% of the total allocated bandwidth for the connection) the bandwidth utilization increased over time, while decreasing the actual bandwidth used per connection. This ensures that the bandwidth is directed to the transmission that actually requires it, rather than on the assumed application type bandwidth requirement (chapter 6).

1.7.5 Resource Harvest Redistribution Call Admission Control Algorithm (RHR-CAC)

Call Admission Control facilitates the end users experience in regards to QoS within the connection oriented paradigm of the IEEE 802.16 standard. This encompasses the requirements of the connection from the initial request through to termination. The standard setup comprises of the SS requesting the required resources and the BS determining if the request can be facilitated, based on the available resources it has to distribute. If the BS can provide the required resources then the connection is accepted and the resources are provisioned for that connection otherwise the request is rejected. The limitations of this process are that firstly resources can be provisioned and not used by the requesting SS. Secondly that many connection requests are rejected due to limited of resources.

The Resource Harvest Redistribution Call Admission Control (RHR-CAC) algorithm prioritises connections fully utilising the available resources of the BS. In turn the algorithm harvests redundant unused resources and then redistributes them to connections which otherwise would be discarded, creating efficiencies in the CAC process. This is achieved while still maintaining adequate QoS of current accepted connections. All the SF QoS categories are considered and balanced rather than focusing on the higher priority SF's and starving the lower SF such as nrtps or BE of connectivity.

Though this algorithm injects additional latency when determining whether to accept a connection, this is balance with increased overall connectivity and more efficient use of finite resources while preventing resource starvation to lower SF categories (see chapter 6).

1.8 Research Methodology

It was Aristotle who made the initial distinction between Science (episteme) and Technology (techne) (Richard, Fall 2008 Edition). In his reflections he saw technology as a continuously changing paradigm, where science is the investigation into the unchangeable, which reflect both the classic and contemporary paradigms of science (Schummer, 2001). The simplicity of Aristotle's distinctive split of the perspective of sciences does not truly reflect the interdisciplinary nature of contemporary sciences (Schummer, 2001). Consequently as technology rapidly evolves the underlying scientific theory remains fairly constant.

The origins of computer science have evolved from a variety of disciplines that interleave their many concepts into abstraction of theory and practice, (Mahoney, 1992), (Savage, 1998). Computer Science is not only about the technology it is also the abstraction of mathematical thinking in finding solutions to problems, while encompassing elements of demanding engineering design skills. Theoretical computer science seeks to understand both the limits of computation and the power of the computational paradigm.

1.8.1 The Scientific Method

Scientists in their quest to discover the solution to questions or problems posed within the science is accomplished by utilising the scientific method where the answer can be formulated as theories or meta-theories. Figure 1.5 illustrates the continuous flow of the scientific change and developments via the Hypothetico-Deductive method (Dodig-Crnkovic, nd). This is propagated by the research community continuously re-examining theories and deducing new-theories or confirming old ones in a new context, therefore the logical nature of science is recursive. From each iteration a hypothesis is predicted which has its origins in the pre-existing body of knowledge from each hypothesis an observation, a theoretical test or practical experiment is derived, which is then placed within the boundary of a certain world view. Critics of the Hypothetical-Deductive method (figure 1.5) argue that there is no such thing as a scientific method (Dodig-Crnkovic, nd; Grimes, 1990). The advantage of this method is that it is impartial; it ensures that experiments can be repeated and results reproduced to enable results to be revalidate or evaluated. Therefore the impartiality of the method demands openness and interoperability. This work requires an iterative approach where incremental experiments are conducted to test the hypothesis.

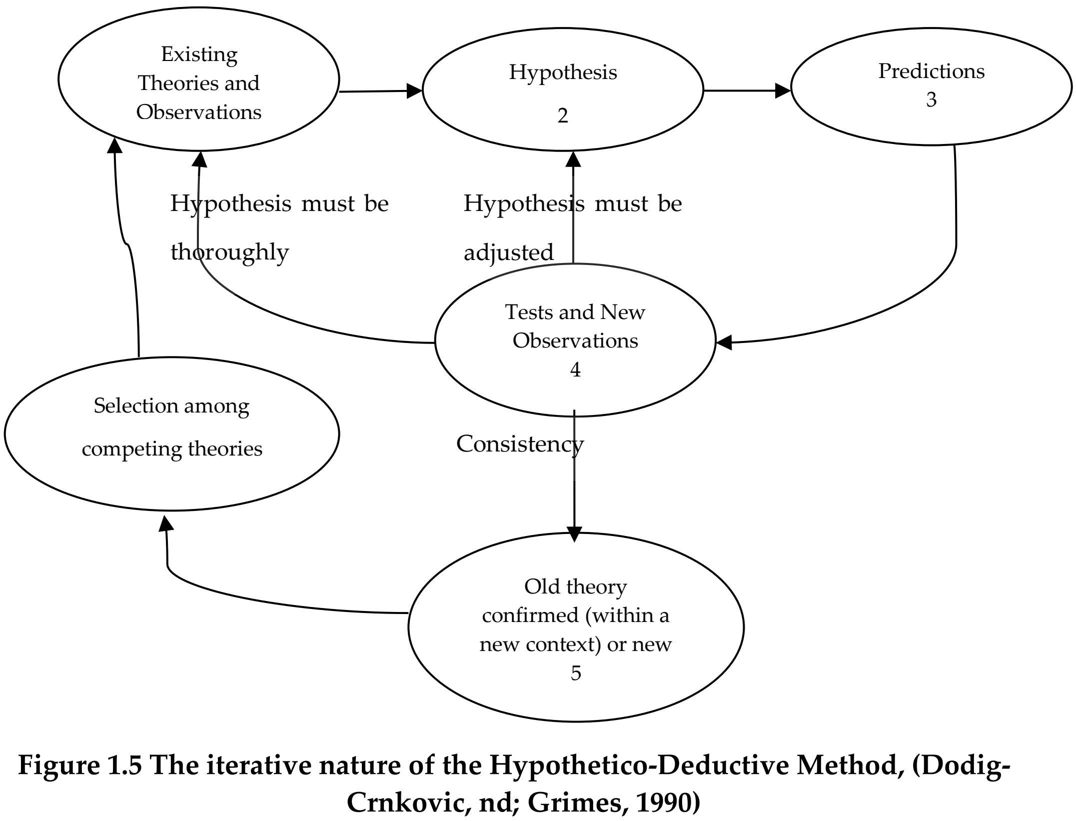

Figure 1.5 The iterative nature of the Hypothetico-Deductive Method, (Dodig-Crnkovic, nd; Grimes, 1990)

1.8.2 Computer Simulation

The challenge of implementing whole system phenomena with the aim of conducting controlled experiments that provide empirical data, enabling improved understanding of the systems parameters within controlled conditions, remains a constraint in the real world. Large scale change in a real-world system is difficult to predict, therefore, some certainty is required before such a change would be implemented.

To study a phenomena it needs to be simplified to enable it to be modelled, to accomplish this, the significant features of the phenomena is identified, and the process encompasses the use of established theory. Once the model has been developed it provides the method for the research to predict consequences of given changes which must be measurable. The reliability of the experiment, is dependent on the ‘quality’ of the model developed. To ensure ‘quality’ the model needs to be comparable to previous models or the real world. Computer simulation provides a ‘method of emulating’ the model required to conduct experiments of the real-world utilizing computational software (Law, 2007).

The use of computer simulation within computer science research complements both theory and experimental methods. It is capable of extending the investigations to study phenomena that can provide the basis for an iterative research approach similar to the hypothetical deductive method. The theory informs the simulations while the computational results can infer new theoretical models. Therefore simulation is an appropriate method to model the complexity of non-linear systems which encompasses the simulation of the real world as evidenced in virtual reality (Ringland, 2010). The simulation methodology is subdivided into three strains, discrete events; system dynamics; and agent-based simulations.

This work utilizes discrete event simulations, where defined systems are decomposed into a set of entities that continually evolve over time within the constraints of the available resources, coupled with triggering events to produce data. Figure 1.6 illustrates an iterative approach to conducting a discrete-event simulation study (Law A. , 2007), (Banks, Carson, Nelson, & Nicol, 2005). This commences by clearly defining the problem to be investigated, with precisely defined boundaries and objectives. The next two sections iterate with collating data (step 2) of the existing system and utilizing data to form assumptions (step 3). The assumptions help to formulate the model to enable it to be replicated in software, and verified (step 4). The model is then tested (step 5) to ensure its validity (step 6) before experiments are designed (step 7) to fulfil the objectives of the study. From this point the simulation is run (step 8) and the data is collated and analysed (step 9) before being documented (step 10). If the model was found to be invalid then the method iterates back to the second criteria where the initial data can be reviewed and the model corrected and validated, if necessary, before progressing through the stages of the method again.

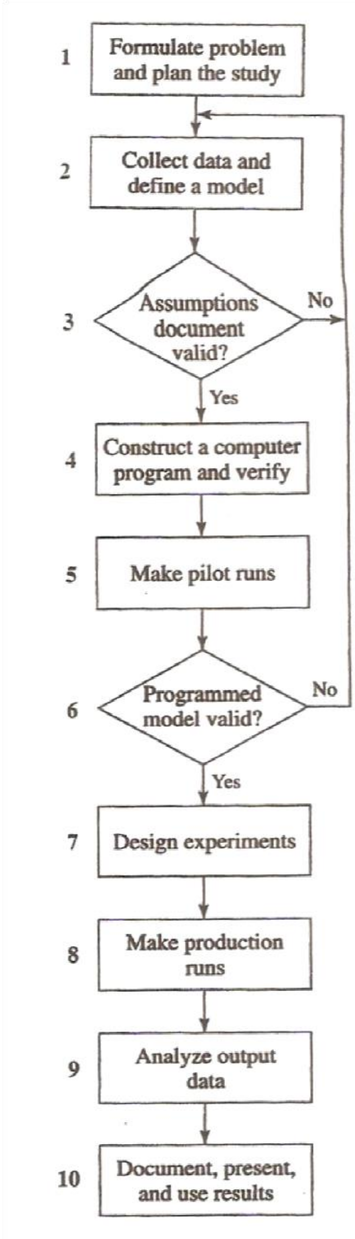

Figure 1.6 Steps in a discrete simulation study (Law A. M., 2003), (Law A. , 2007), (Banks, Carson, Nelson, & Nicol, 2005)

1.8.3 Interleaving of the hypothetico-deductive model and the simulation study methodology

The discrete simulation study methodology illustrated above (figure 1.5) incorporates the Hypothetico-Deductive Method in relation to the scientific method (figure 1.5) that maps to the mind map illustrated in figure 1.7, that details the key stages of this study.

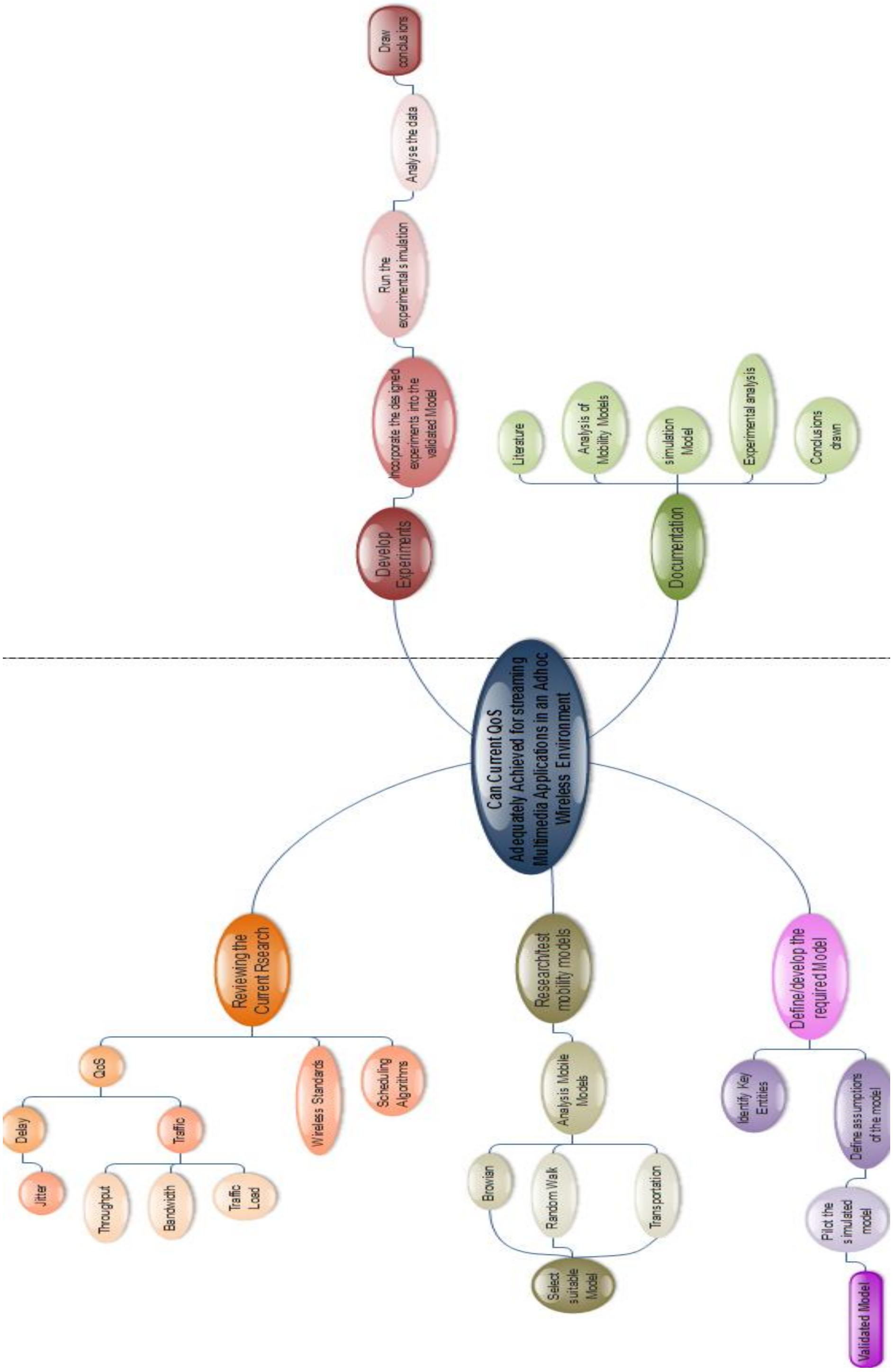

Figure 1.7 mind map of the key stages of the research being undertaken

The mind map (figure 1.7) illustrates how the review of the literature will substantiate the hypothesis, for this work, while eliciting the key entities for the core simulated model. Together with the review of the literature which incorporates the field of QoS within the IEEE 802.16 standard and the investigation into mobility models such as the Brownian, random walk and transportation models determining the mobility requirements of the ad hoc network that will be incorporated within the core simulation. With the aim to replicate the real world roaming pattern of the users' connectivity to a WiMAX network. Once the key entities and the mobility model have been ascertained then the simulation can be designed and the model validated. The validation process compares actual data from the simulation to expected outputs. From this point the experiments can be designed to test the hypothesis. Therefore the experimental design and development will take an iterative incremental approach testing one variable at a time to establish the impact on the overall model to ensure that the data output do not become corrupted. The simulations will be run a number of times to ensure consistent and replicable data using the method of independent replication (Law A. , 2007). The data will then be analysed and conclusions drawn, to test the validity of the hypothesis, ascertaining that QoS can be adequately achieved for streaming multimedia applications in an ad hoc wireless network; combining the practicality of the discrete simulation methodology data to infer theories that establishes the hypothesis.

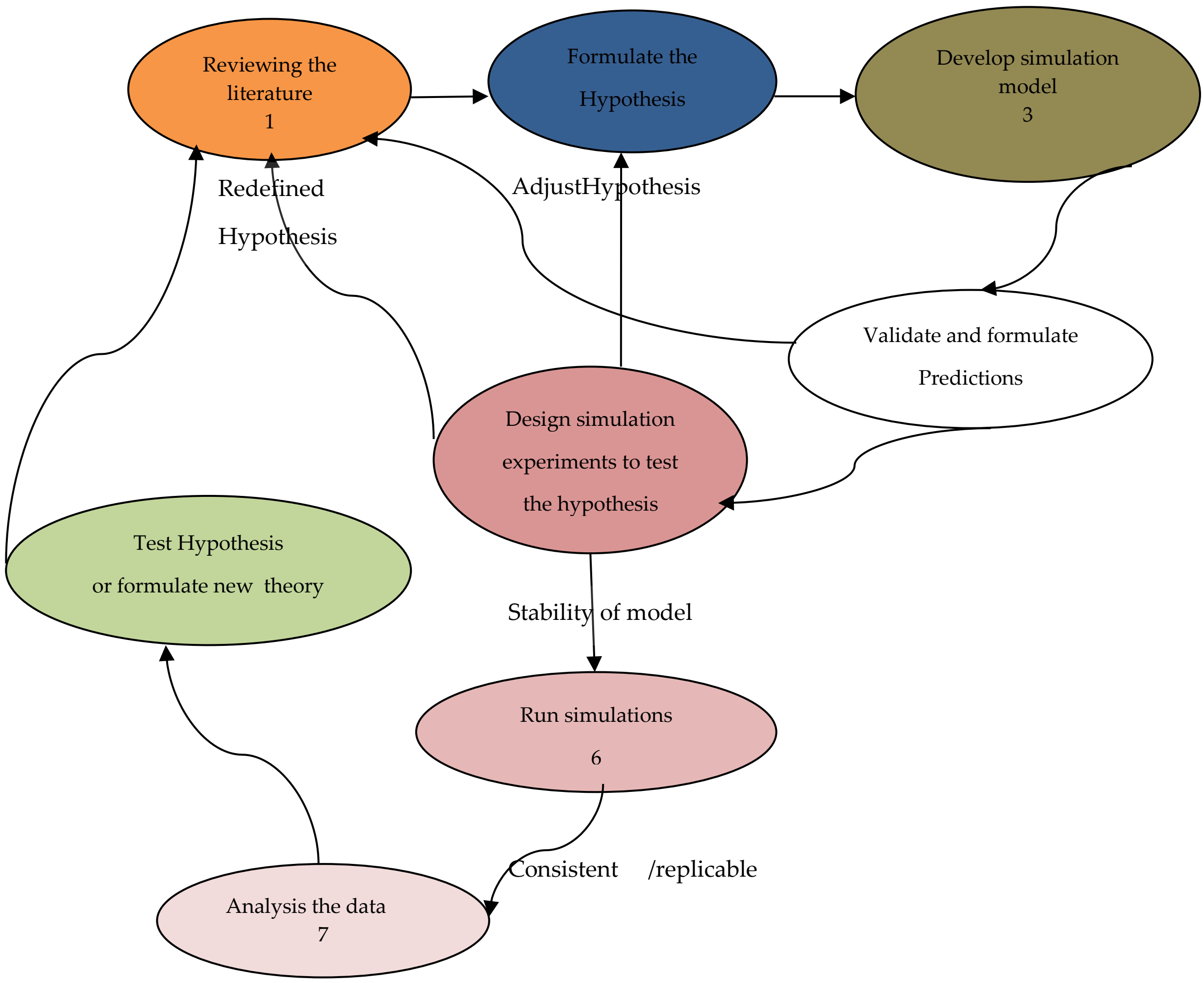

Figure 1.8 depicts a conceptual model of how the key features of the simulation study model, Hypothetico-Deductive Method and the mind map

Figure 1.8 illustrates the interleaving of the key research stages, depicted with colour that correlate to each stage in the mind map, figure 1.7, together with the simulation study model and hypothetico-deductive method highlights the iterative process of the scientific investigation and research methodology of this thesis.

Summary

The human need to communication instantly, unambiguously with clarity and no physical boundaries still remains a significant basic need today, driving the necessity to exchange information and in turn build communities.

Communication was originally limited to the physical infrastructure but radio changed this and delivered mass affordable communication which enabled cross continental communication as the need to communicate grew this development became more important. With ALOHANET providing the first WLAN that connected to ARPANET as a backbone but even today the QoS of the wireless environment still presents many challenges. We experience the flexibility of instant pushed data. to our devices, the mobility of the message is still paramount together with the quality of the delivery. But still, the challenges to achieve this with finite resources remain, this work will investigate, if in the WiMAX domain, can QoS still be achieved coupled with the ever increasing demand from multimedia applications.

The strong correlation between computer science and technology indicates the critical mass of scientific features that are encompassed within contemporary science. "*Simulation is a powerful research method that enables researchers to look at an artificial world move forward into the future, giving the user the unprecedented opportunity to intervene and attempt to make improvements to performance*", (Law A. , 2007). The epistemological approach will follow the interleaving of the Hypothetico-Deductive Method and the discrete simulation methodology provides the approach to test and analysis the hypothesis of this work. The key elements of this study (see figure 1.16) have also been incorporated within the research methodology providing an iterative approach of deducing theory via simulations to infer the improvement of future systems.

The next chapter will investigate the challenges of providing QoS within the wireless environment to the end user. In particular the work will focus on the IEEE 802.16 standard, after providing an overview of competing communication technologies, in which a framework to the key variables contributing to the outcomes of this work are investigated demonstrating the necessity of good design utilising efficient algorithms to enhance the quality of the user experience. To ascertain good QoS key aspects of both scheduling and call admissions control that influences the quality within IEEE 802.16 communication.

This thesis follows the iterative approach to proving the hypothesis. Chapter 2 reviews the domain of wireless technology and in turn defines the concept of QoS. Both the scheduling and call admissions control aspects of the WiMAX architecture defines the theory for the two experiments in chapter 5. These tested the efficient use of resources from two perspectives. These results informed the RHR=CAC algorithm, which is discussed in chapter 5. Chapter 6 concludes the success of this research and identifies future work.

Chapter 2 Quality of Service within the Wireless Domain

This chapter introduces QoS with respect to the wireless networked environment, through an appreciation of the intricacies of delivering QoS to the end user, together with the current protocols and policies that aim to guarantee that QoS is adequate. The key standards that influence wireless connectivity are the IEEE 802.11 and IEEE 802.16, both these standards utilise the MAC layer functionality to provide QoS management for network traffic, that encompass mechanisms and parameters used to achieve the levels of service quality that today's users' experience and expect. Therefore this chapter will address QoS mechanisms within both the IEEE 802.11 and 802.16 standards. The primary, focus is on the IEEE 802.16 standard which provides broadband level wireless throughput incorporating the potential for improved mechanisms to enhance QoS while highlighting the challenges associated with this. John Ruskin (1819 – 1900) stated *"Quality is never an accident, it is always the result of high intention, sincere effort, intelligent direction and skilful execution; it represents the wise choice of many alternatives"* (Ruskin, 1819 - 1900). This is also true for wireless network communication, only good design and implementation along with the appropriate algorithmic combination of mechanisms and parameters will enhance the user experience. Initially an overview of the technology will be presented after which the focus will switch to investigating scheduling particularly the impact on QoS. The chapter then concludes with an investigation into advances with QoS within the IEEE 802.16 paradigm

2.1 Introduction to Quality of Service

Today's, networks need to support the demands of variable traffic types over converging network infrastructures. Each application requires a differentiated service to ensure that it is received in a functional state. The demand for a reliable network infrastructure instigated the need for QoS in commercial networks, thereby promoting the development of a standard with the aim to deliver an adequate quality performance to the end user (ITU, 2004). In the ITU-T recommendation E.800, QoS is "*the collective effect of service performances, which determine the degree of satisfaction of a user*" (ITU-T Recommendation E.800, 2008). From the commercial perspective Microsoft (2011) defines QoS as "*...as set of technologies for managing network traffic in a cost effective manner to enhance the user experience of the service in home and enterprise environments*" Microsoft goes on to define QoS still further from the technological view "*...QoS technologies allow you to measure bandwidth, detect changing network conditions (such as congestion or availability of bandwidth) and prioritize or throttle traffic....*" (Microsoft, 2011). The ITU-T views QoS from a humanistic perspective whereas Microsoft's view is very mechanistic. This work will take advantage of the mechanistic elements of the IEEE 802.16 communication standard by providing a means to manage the efficient distribution of resources by utilizing the defined parameters while maintaining the importance of improving the humanistic experience of the technology.

Traditionally the best effort was the service mode for networks where network traffic is treated fairly with no guarantees of performance (Floyd & Allman, 2008). The constraint of this model is, when the network is running a bandwidth hungry application, such as streaming multimedia, VoIP, IP-TV and online games, it is assumed that all other applications can suffer by not receiving adequate bandwidth to transmit. To accommodate this situation if an application is mission-critical in a commercial environment certain traffic types may, then need preferential or priority treatment within the whole system to fulfil those mission-critical requirements. Therefore the aim of QoS is to balance the network resources with the network delivery requirements, thus providing an overall satisfactory network performance. In achieving this bandwidth, latency, latency variation commonly known as jitter and data throughput need to be managed efficiently (Microsoft, 2011). Figure 2.1 illustrates the latency in the connection of Page Load Times (PLT) based on users in the US with an average bandwidth of 5mps. It is apparent that the increase in bandwidth from 1mbps to 2 mbps halves the latency but increases in bandwidth thereafter do not produce such significant gains, as can be seem between 3 -10mbps (Grigorik, 2012). Therefore inferred from this data, it is not the amount of bandwidth that produces good QoS but efficiencies of the algorithms used to aid the throughput of the applications utilizing the bandwidth (Franciszek, Paszkiewicz, & Bolanowski, 2013).

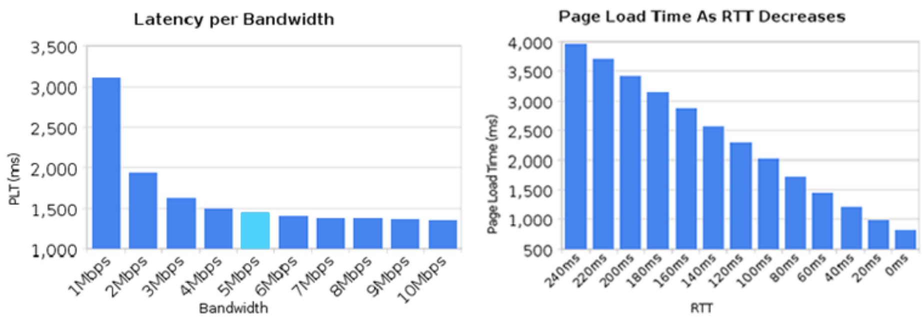

Figure 2.1 Latency: The New Web Performance Bottleneck (Grigorik, 2012)

QoS mechanisms can also be used to manage protocols such as the User Datagram Protocol (UDP) which are inherently unreliable as they do not deploy acknowledgements and therefore cannot detect network congestion (Tanenbaum & Wetherall, 2010). Therefore, QoS mechanisms can balance resource distribution by allowing the mission critical application priority access to their required resources, while still allowing other applications access to enough resources to function. It is the Call Admission Control mechanism (see chapter 4) that decides how, when, and who will be allocated available network resources. The traffic control mechanism manages the data flows by classifying, scheduling and marking packets based on priorities by shaping traffic. Together they can categorize traffic into service classes and control delivery to the network based on these service classes.

2.2 Network Technologies that Support QoS

QoS is dependent on all the components throughout the network supporting the standards and mechanisms enforced. Sir Tim Berners-Lee, cited by Cellan-Jones (2008), believes that accessibility to the internet is all about open standards and interoperability (Cellan-Jones, 2008). The risk is, if a network device en-route does not support QoS standards, the traffic flow will resort to First Come First Serve (FCFS) or best effort SF at that point, compromising any attempt to provide QoS, emphasizing why compliance to open standards are vital.

	Best Effort	DiffServ	IntServ
Services	• Connectivity • No Isolation • No Guarantee	• Per aggregated isolation • Per aggregated guarantee	• Per flow isolation • Per flow guarantee
Service Scope	• End-to-end	• Domain	• End-to-end
Complexity	• No set-up	• Long term setup	• Per flow set up
Scalability	• Highly Scalable • Nodes maintain only the routing state	• Scalable • Edge routers maintain per aggregated state whereas core routers per class state	• Not Suitable • Each router maintains per flow state.

Table 2.1 Comparison of Best Effort, DiffServ and IntServ service models (adapted from (Neelakanta, 2000)).

Two architectures that further support QoS are firstly an Integrated Service (IntServ) which is a fine grained QoS system and secondly Differentiated Services (DiffServ) which is a course grained QoS architecture (Huston, 2000). Whilst IntServ supports individual applications such as VoIP, the throughput is based on mathematical calculated guarantees which may differ from the actual requirements in some cases (Soldatos, Vayias, & Kormentzas, 2005) Table 2.1 illustrates a comparison between best effort, DiffServ and IntServ . The issue with IntServ is that it is complex and therefore limits the scalability of the communication system. Whereas DiffServ offers a more flexible service model than IntServ, as the communication providers can set their own priorities that will integrate into the core of the network. The key factor to ensuring DiffServ performs as expected, is to couple it with bandwidth management controls to improve throughput guarantees (Neelakanta, 2000), neither provides an infallible performance in isolation.

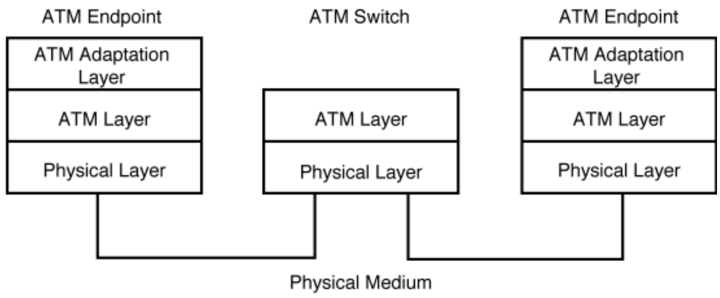

Figure 2.2 ATM Protocol Reference model (Held, 1999)

The Asynchronous Transfer Mode (ATM) architecture , see figure 2.2 (Held, 1999) incorporates QoS with consideration of per flow requirements which were driven by perceived application needs categorised into flows for differentiated QoS support (ITU-T Recommendation I.150, 1999). These developments within ATM have promoted the shift from Best Effort to a differentiated service, encompassing flow aggregation, primarily a fixed wired infrastructure ATM still sets the standard for QoS in any network environment.

Unfortunately, service level guarantees and resource allocation such as those provided by ATM are not currently supported by the more traditional IEEE 802.x LAN wireless technologies at the hardware level (Microsoft, 2011). But QoS mechanisms can be implemented via the higher levels of the open systems interconnection reference model, such as the MAC layer (Held, 1999). The IEEE 802.11 standard is based on the connectionless protocol which is inherently challenging to establish the current state of the network, making resource allocation difficult, as the traffic is subject to delay at nondeterministic points making bandwidth availability and delivery times unpredictable. ATM can predict the key parameters as it functions in a connection oriented paradigm within the wired domain in a similar way to the IEEE 802.16 standard.

2.3 IEEE 802.11: The Wireless QoS Evolution

Standard	Released Date	Data Rate	Modulation Scheme
IEEE 802.11	June 1997	Up to 2Mbps in the 2.4GHz band	FHSS or DSSS
IEEE 802.11a	September 1999	Up to 54Mbps in the 2.4GHz band	OFDM
IEEE 802.11b	September 1999	Up to 11Mbps in the 2.4GHz band	DSS with CCK
IEEE 802.11e	2005	Up to 54Mbps in the 2.4GHz band	TDMA/OFDM
IEEE 802.11g	June 2003	Up to 54Mbps in the 2.4GHz band	OFDM above 20Mbps, DSSS with CCK below 20Mbps
IEEE 802.11n	Sept 2009	Up to 54Mbps in the 2.4 or 5GHz band	CCK,DSSS OR OFDM Can utilise MIMO
IEEE 802.11ac	Jan 2014	Up to 600Mbps in the 5 to 6.3 GHz band	BPSK,QPSK 16 – 256 QAM

Table 2.2 The Evolution of the IEEE 802.11 standard (IEEE, 1997), (IEEE, 2003), (IEEE Std 802.11e-2005, 2005) (IEEE 802, 2014; IEEE, 2009).

Since the initial inception of the IEEE 802.11 standard of WiFi in 1997 the standard continues to advance with each version incorporating additional features for the PHY and MAC layers, see table 2.2, (IEEE, 1997), (IEEE, 2003), (IEEE WG802.11 - Wireless LAN Working Group, 2005), (IEEE, 2009), (IEEE 802, 2014). These features have evolved from the past standards of low bandwidth wireless communication system capabilities to the emergence of a higher bandwidth infrastructure that is now capable of transmitting bandwidth hungry, time sensitive multimedia applications, such as IP-TV. The standard transmits on 2.4-2.485GHz and 5.725-5.85 GHz, the higher frequencies provide more channels, with reduced interference along with enhanced throughput (IEEE WG802.11 - Wireless LAN Working Group, 2005). Within the family of IEEE 802.11 standards, QoS had not been a consideration until the IEEE 802.11e version was ratified in 2005 (IEEE WG802.11 - Wireless LAN Working Group, 2005). This version of the standard provisioned the ability to configure the network to facilitate the demands of multimedia applications. This encompasse prioritization of data, voice and video transmission which is achieved by utilizing the MAC layer, along with Time Division Multiple Access (TDMA) and an error correcting mechanisms for delay sensitive applications.

The standard encompasses two access mechanisms. Firstly the Distributed Coordination Function (DCF) that is embedded within the MAC layer and uses CSMA/CA coupled with a backoff algorithm (IEEE WG802.11 - Wireless LAN Working Group, 2005). In a multiuser environment many stations may try to transmit at the same time. They may all have found the medium idle therefore all stations initiate a transmission which has a high probability of resulting in a collision. Invoking the DCF mechanisms mean that if the medium is found to be idle a backoff algorithm is initiated which delays the station's transmission. The back off time is defined as:

$$\textbf{BackoffTime = Randon() X SlotTime}. \qquad (2.1)$$

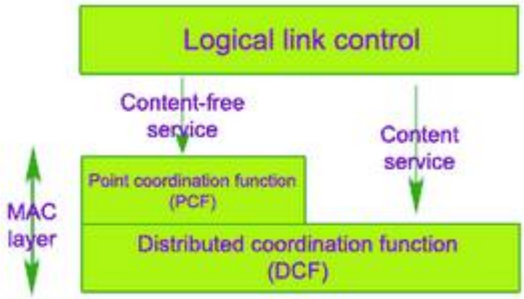

Figure 2.3 Point Coordination Function and Distributed Coordination Function Architecture (Rodrigues, 2006)

This will prevent many collisions but to avoid them totally the method also utilizes the CSMA/CA, which encompasses Request-To-Send (RTS) and Clear-To-Send (CTS) protocols to ensure the medium is ready to receive the stations transmission. To further ensure a successful transmission and in turn improve the quality of the transmission via the second access mechanism. Point Coordination Function (PCF) polls stations from the access point (AP) from here it can coordinate channel access proirity (IEEE WG802.11 - Wireless LAN Working Group, 2005). This is achieved by gaining control of the channel through the PCF Interframe Space (PIFS) duration allowing priority access to the channel as it sits above DCF within the architecture (see figure 2.3), which makes PCF ideal for controlling multimedia transmissions. The PCF mechanism can suffer from the hidden station problem where a station (see figure 2.4) that communicates directly with the AP may not be aware of others around it, especially if other stations are at the geographical edge of the transmission range this is due to attenuation of the signal.

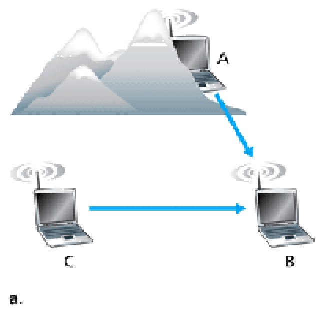

Figure 2.4 Hidden Station Problem - Hosts may be within range of the AP, but not each other -- they are hidden from each other.

To improve QoS provisioning the IEEE 802.11e version of the standard incorporates Enhance Distributed Coordination Function (EDCF), which provides variables CWmin and CWmax to determine a random backoff value based on traffic classifications. The traffic classifications are priority based and broken down into eight classifications which are then channelled into four access categories figure 2.5 (IEEE Std 802.11e-2005, 2005). The station sends their packets once they detect the medium is idle, they then wait an Artibitration Interframe Space (AIFS). The AIFS are variable sizes and higher priority traffic have shorter periods of time in AIFS. Collisions are also avoided by the Contention Window (CW), which is where a station counts a random number of times slots before transmitting. If another station commences its transmission before the countdown is complete the countdown of the initial station starts again. Within the IEEE 802.16's connection oriented paradigm the overheads of these protocols are negated by the point to point connection, coupled with the CAC procedure.

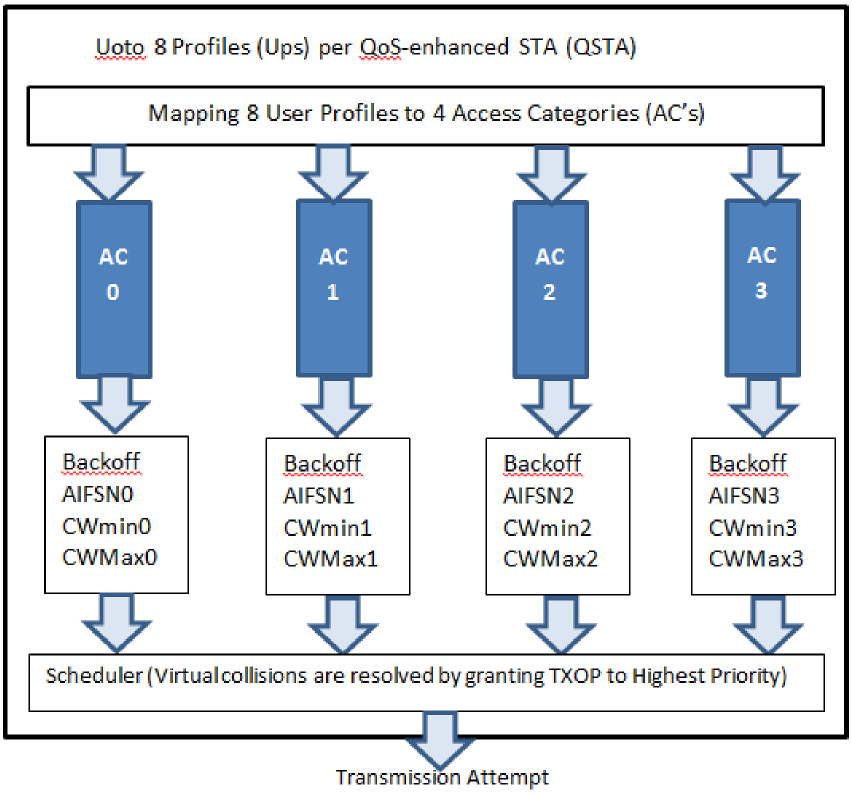

Figure 2.5 Mapping in EDCF in IEEE 802.11e (IEEE Std 802.11e-2005, 2005)

The standard also extends the polling mechanisms via the Hybrid Coordination Function (HCF), which allocates a start time and a maximum transmit duration to promote QoS (Shepard, 2004). A hybrid controller polls stations during a contention-free period. One node employing HCF is responsible for the management of the wireless medium via HCF Controlled Channel Access (HCCA). It is able to allocate Transmission Opportunity (TXOP) to itself after a gap of PIFS, resulting in a priority over other EDCF nodes that cannot access the wireless medium until the completion of AIFS_DIFS_PIFS. HCF has two TXOP's methods, the first being EDCF-TXOP for contention using wireless medium access via EDCF methods. The second method is polled-TXOP, where EDCF-TXOP's can be obtained only in the contention-period (CP) while polled-TXOP's can be allocated by the HCF via

Enhanced Distributed Channel Access (EDCA) (Wiethőlter & Hoene, 2003). This enables the HCF to take control by sending a QoS CF-poll frame after a PIFS-idle medium (eliminating the backoff). The TXOP's in the contention-free period (CFP) are allocated by HCF with CF-poll frames including the starting point and the maximum duration. A CF-end frame of the HCF or the point of time announced in the beacon frame leads to the end of CFP. Though this method provides a priority based mechanism for bandwidth allocation, it still does not provide guarantees of service for the entire transmission, QoS becomes inherently complex over a connectionless environment, though the idea of priorities are similar to that detailed within the IEEE 802.16 standard.

To further enhance the HCF polling mechanism a comparative experiment was conducted to test the efficiency of HCF within a noisy wireless domain utilizing the IEEE 802.11 standard. The focus was to further improve the QoS for high priority traffic, this incorporated fragment-to-fragment acknowledgements together with a proposed block-frame acknowledgement. It was found that the block-frame acknowledges proposal demonstrated improvements in efficiency and in turn QoS, though a more dynamic algorithm would have further enhanced the results (Adda & Peart , 2006).

The connectionless nature of the IEEE 802.11e standard means achieving end to end transmission with guarantees of service quality is still a challenge even with the mechanisms that have been implemented. The fact that the transmission will still rely on other protocols that do not encompass QoS such as UDP means that true QoS still eludes the IEEE 802.11 standard. To ensure the required QoS from end to end ideally a connection oriented environment would need to be deployed such as incorporated in the IEEE 802.16 standard with greater bandwidth capacity, while still encompassing the priority categories of application traffic.

2.4 IEEE 802.16 Standard: The Quality of Service Perspective

Cisco networking, defines QoS as the "*Capability of a network to provide better service to selected network traffic over various technologies*" (Cisco networking, 2003). Whereas the ITU-T X.902, (2009), recommendation defines QoS as "*A set of quality requirements on the collective behaviour of one or more objects*" (ITU-T Study Group 17, 2009). Cisco and the ITU-T view QoS from different perspectives, Cisco's approach to QoS is determined by the physical network to provide a service, whereas the ITU-T defines quality as the requirements of the object, this could be the needs of an application such as VoIP to function across a network satisfactorily (Tung, Tsang, Lee, & Ko, 2008). This is a different definition to the ITU-T E.800 recommendation and Microsoft that differentiated quality from both the humanistic and mechanistic perspectives (see section 2.1 Introducing QoS) where the ITU-T highlighted the user experience as the measure of quality, which is more difficult to ascertain. A significant advantage to the IEEE 802.16 standard is it uses a connection oriented protocol with multiple connections capable of differentiated SFs. Therefore it has the potential to deliver a QoS communication system from both the infrastructure and user perspective.

An alternative view of QoS is the Grade of Service (GoS) this incorporates connection capacity and network coverage which includes guaranteed maximum blocking and outage probability, this perspective is adopted from the telecommunications industry (ITU-T study group 2, nd). IEEE 802.16 is designed to work with many data types e.g. streamed multimedia, VoIP as well as traditional data transmission, with these applications users demand a satisfactory level of service to maintain the usability of the application. The standard aims to provide more bandwidth to the right channels at the right time reducing latency and jitter (Ohrtman, 2005), making it ideal for time sensitive applications. Applications that are distributed across a network can become unusable if there is no QoS protocol available. The ITU-T (ITU-T G.114), specifies latency should be less than 150ms to achieve a satisfactory QoS in regards to this parameter (International Telecommunication Union - Telecommunication, 1996). The goals of QoS mechanisms are to reduce key variables that influence the quality of the end user experience, such as latency, jitter and packet loss while providing adequate bandwidth.

For example, to achieve a quality service for VoIP, latency should not exceed 150ms, hence the ITU-T requirement, for a one way speaker to listener scenario, while the voice quality deteriorates to such an extent that the human ear finds it difficult to decipher the actual words transmitted. The human ear can normally cope with latency up to 250ms (ITU-T 2005). Once the latency reaches 200ms, QoS is reduced to such an extent, the human ear can no longer decipher the actual words clearly. But even if the latency is as little as 65ms or 30db of attenuation, an echo will be produced that will render the application unusable (see figure 2.6) (VoIPforo.com, nd). IEEE 802.16 standard has strived to integrated control of the key parameters of QoS to provide a satisfactory experience for the end user. This is achieved by utilizing multiple connections per channel providing efficient throughput for a variety of differing traffic models (Ghosh, Wolter, Andrew, & Chen, 2005). QoS provisioning is handled by the middle layers of the OSI reference model such as the MAC layer.

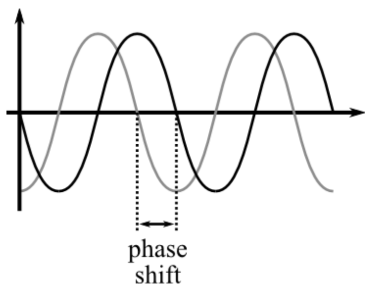

Figure 2.6 An Example of phase delay that if excessive will render an audio application useable

2.5 Evolution of the IEEE 802.16 Standard

The IEEE 802.16 standard commonly referred to as WiMAX, and used interchangeably throughout this work, incorporates both fixed and mobile Internet access, (see table 2.3). The WiMAX forum controls the certification so as to ensure WiMAX enabled devices are compatible with the standard's requirements, such as the Samsung SWD-M100 Mondi (see figure 2.7) which was the first WIMAX enabled device compatible with the Clearwire distribution of WiMAX in the United States (Samsung, 2009). Also in 2010, HTC produced a WiMAX ready handset with the 'HTC A9292 WiMAX bar' (see figure 2.8) this device was compatible with Sprint, again focused at the US market (Nickinson, 2010). The 2012 evolution of the standard saw WiMAX ready device expand to in excess of fifteen or more predominately from HTC, Blackberry and Samsung. The Standard offers throughput of between 40Mbps (which is less than the current IEEE 802.11 standard) through to 1Gbps on a fixed infrastructure which the mobile domain can reach data rates of 75mbps. The aim of WiMAX is *"a standard-based technology enabling the delivery of '**last mile**' wireless broadband access as an alternative to cable and DSL"* (Wimax Forum, 2011).

IEEE Standard	Description Key Components
802.16	Initial ratification of the standard including line of sight (LOS), Point-to-Multi-Point (PMP) wireless access applications using 10-66GHz Spectrum.
802.16a	This amendment introduced the spectrum of 2-11GHz for Non-Line-of-Sight (NLOS), NLOS can reach a range of 4 miles. The mesh topology and OFDM was also incorporated in this revision.
802.16b	Further QoS provisioning was integrated
802.16-2004	The whole standard saw a major revision and the previous amendments being amalgamated into this revision and completed the fixed wireless standard
802.16e-2005	QoS and Mobility features was incorporated into this amendment, incorporating mobile communication at vehicular speed of approximately 60km/h
802.16-2012	Air Interface for Broadband Wireless Access systems Rollup of 802.16h, 802.16j and 802.16m standards WirelessMAN-Advanced Air Interface for Broadband Wireless Access Systems

Table 2.3 Evolution of the IEEE 802.16 Standard (IEEE, 2011; IEEE , 2012)

To accomplish this, a software implemented gateway technology called Softswitch is utilised, (International Softswitch Consortium, 2008). This provides a bridge from the PSTN switched infrastructure to the IP network making a viable alternative for voice communication. When delivered in the PMP topology distances of up to 30 miles can be reached. It is envisaged that by utilizing WiMAX last mile backhaul bypass capabilities will make it ideal for places such as rural Britain who struggle to gain QoS Internet connectivity.

2.6 Quality of Service Mechanisms built into the IEEE 802.16

There are some similarities and influences on the PHY layer design in IEEE 802.16e from the IEEE 802.11a standard. Though these two technologies work in different paradigms, in that their key fundamental protocols differ, they both still utilize OFDM. The key distinguishing features of the IEEE 802.16 standard is that Orthogonal Frequency Division Multiplexing Access (OFDMA) and OFDM subcarriers can then be adaptively modulated depending on the distance and noise of the transmission within the bandwidth. OFDM increases the spectral efficiency and an improved handling of the interferences that are also incorporated into OFDMA which provide a scalable option that provides higher efficiency in bandwidth use. Originally the standard was developed to work with a point-to-multi-point topology. Table 2.4 defines the modulation and coding types which are used in the QoS mechanisms within this standard.

Modulation Type	Description
TDD	Dynamically allocate uplink and downlink bandwidth based on requirements. This has 1 uplink and 1 downlink subframe separated by a guard slot. The number of slots are adaptively allocated according to the bandwidth needs
FDD	BS transmit on different sub-bands eliminating interference allows for bandwidth allocation flexibility
OFDM	Greater spectral efficiency and migrating interference with its tighter beam width and its dispersal at data across different frequencies **Forward Error Correction FEC** –builds redundancy into the transmission by repeating some of the information bits, which helps lost and missing bits at the receiving end, this prevents whole frame requiring retransmission in turn this would increase delay and impact on QoS. The data can be carried of a number of sub carriers. If error on one sub carriers the than data can still be received at the destination as the data is interleaved on the subcarrier by placing the bits close together in time but spread across the frequency. Therefore if there are errors on an individual subcarrier the error would be a small part of the data and therefore easier to be correct with FEC. Fast Fairer Transforms (FFT) is a discrete Fairer transformation theorem, transforming the time domain into the frequency domain (hyperphysics, 2011). It provides efficiency in evaluating complex numbers, by greatly reducing the number of computational operations required. A discrete Fourier Transform takes $O(N^2)$ computational operational while FFT can compute the same outcome in O(N LogN) operations. If computing large

	data sets than the speed efficiencies provided by this algorithm is of huge importance in improving QoS. OFDM converts a single data stream into M streams and modulates them into M subcarriers using FFT.
OFDMA	Used for Mobile WiMAX primarily, multiple closely spaced sub-carriers which are then divided into groups of sub carriers that create sub-channel. Non-adjacent sub-carriers can be included in the same sub-channel. It provides multiplexing operations of data streams from multiple users onto the downlink and uplink by means of sub-channels. The sub-channels form the minimum frequency resource-unit allocated by the BS. Different sub-channelization's maybe allocated to different users as a multiple-access mechanism. Two of the subcarrier schemes use in mobile WiMAX is Partial Usage of Subcarriers (PUSC) this uses 15 and 17 channels for both the uplink and downlink on 5MHz, this is 30 and 35 channels for 10 MHz. Adaptive Modulation and Coding (AMC), here sub-channels are allocated based on user frequency. System capacity can be improved if the sub-channels maximizing its SINR

Table 2.4 Modulation and Coding Explained (Andrews, Ghosh, & Muhamed, 2007)

2.7 PHY Layer: Modulation

Quadrature Amplitude Modulation (QAM), is a function embedded into the PHY layer of the WiMAX architecture and recently incorporated into the IEEE 802.11ac standard (table 2.2). QAM is used for modulating data signals onto a carrier for radio communications. The QAM signal utilizes two carriers, shifted in phase by 90 degrees and that are modulated producing both amplitude and phase modulations (Radio-Electronics, 2011). QAM is able to carry higher data rates than ordinary amplitude schemes. The higher order modulation formats, manipulate more constellation points (see figure 2.7) to transmit more bits per symbol. There are some disadvantages of QAM: for example one is if the constellation points are close together they are more susceptible to noise and data error, if QAM uses an amplitude component, a linear amplifier must be used, these are less efficient and consume more power causing problems for mobile device battery usage (Radio-Electronics, 2011).

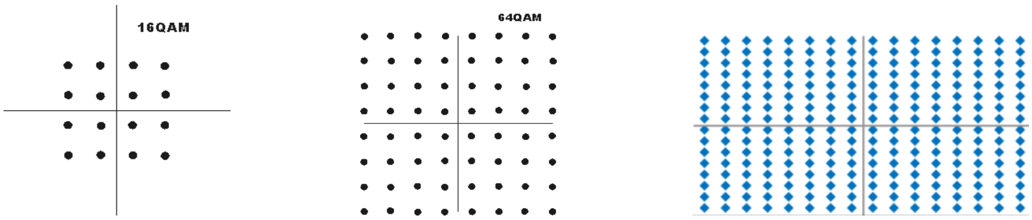

Figure 2.7 Comparison of 16-QAM, 64-QAM and 256 QAM

2.7.1 Adaptive Modulation

Adaptive modulation enables a dynamic bandwidth allocation to match the current channel conditions. Higher order modulation algorithms are more vulnerable to interference and noise, but are able to sustain a higher throughput. The compromise is balancing the frequency or spectral range of the modulation and the throughput (Pelcat, 2013).

The Base Station (BS) monitors the connection explicitly via the distance and the current atmospheric conditions (table 2.5). If the BS finds difficulty connecting to a SS using 64-QAM it will reduce the modulation to 16-QAM or further to QPSK dynamically, if the conditions require this. The 16 QAM is more robust than 64-QAM but the concession is a lower data rate. As can be inferred from both figure 2.8 and table 2.5 by reducing the modulation type the throughput and range are further constrained.

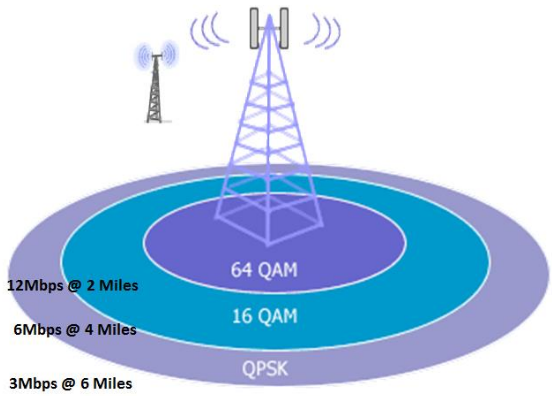

Figure 2.8 WiMAX modulation and coding schemes illustrating throughput and distance parameters.

Modulation	WiMAX Throughput and Range	Bits per symbol	Error margin		Complexity
QPSK	3Mbps @ 6 Miles	1	1 / √2	0.71	Medium
16 QAM	6Mbps @ 4 Miles	4	√2 / 6	0.23	High
64QAM	12Mbps @ 2 Miles	6	&radic / 14	0.1	High

Table 2.5 Summary of types of modulation with data capacities (Radio-Electronics, 2011)

2.7.2 Adaptive Burst Profiles

Adaptive Burst Profiling provides a mechanism for QoS requirements to be met under various channel conditions. Adaptive Burst Profiling allows for the transmission parameters, including modulation type, Forward Error Correction (FEC), preamble length and guard times, to be adjusted individually to the needs of each SS on a frame-by-frame basis according to the link conditions (figure 2.6). Different combinations of the FEC and modulation provide various levels of QoS assurances. The TDD and FDD configuration's both support adaptive burst profiling on both the uplink and downlink channels. In the uplink the SS transmits in a given time slot with a specific burst size. The downlink transmissions to the SS can also be aggregated into the same downlink burst.

High order modulation	Higher transmission speed, susceptible to interference
Low order modulation	Lower transmission speed, can cover greater distances
FEC	Reduces retransmissions, coding dose require additional bits and therefore injects redundancy.

Table 2.6 Modulation and FEC Profiling

2.8 IEEE 802.16: QoS Provisioning

The IEEE 802.16 connection is assigned a 16 bit unique Connection ID (CID) (see section 2.11.1 on Convergence sublayer) which correlates to a Service Flow ID (SFID), where both are then allocated a service class to deliver the QoS parameters required. This is initiated when an application registers with the network. It is the convergence sublayer within the MAC layer that maps the data into a QoS class. Built within this layer is a queuing and traffic shaping engine which is responsible for the transmission and receiving of the packets. By utilizing CID's the WiMAX standard is able to deploy connection oriented network protocol over the Internet Protocol (IP). The ability to associate each packet with a SF is what distinguishes IEEE 802.16 from the IEEE 802.11, 3G and 4G standards (Haseeb & Tralli, 2011).

2.9 Service Flow Categorizations

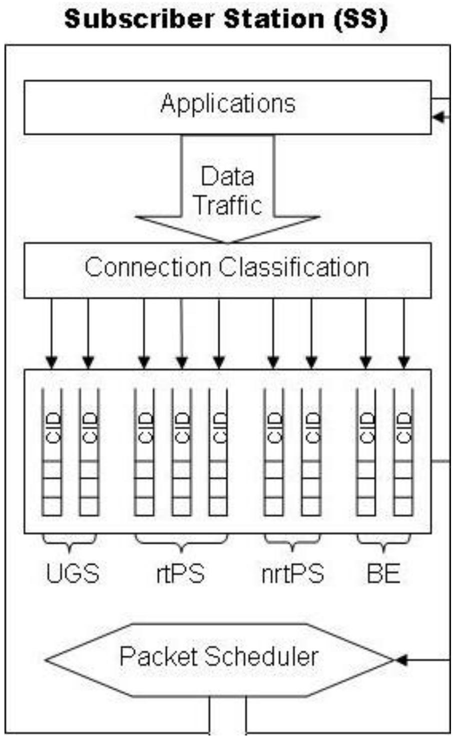

Figure 2.9 Architecture of the Subscriber Station Illustrating the Scheduling Services (IEEE 802.16e-2005, IEEE Std 802.16-2004/Cor1-2005, 2006)

The strength of the IEEE 802.16 standard is that it ensures that QoS requirements are met as reliably as possible. This is accomplished in part by the standard's implementation of scheduling SF, these categorizes the QoS requirements of different network traffic types. Within the traffic types the traffic is categorized according to different performance indicators that include throughput, latency, jitter and packet error rate tolerance, which are encapsulated in the QoS parameters. Below are details of the five categories encompassed in the standard. Figure 2.9 illustrates the SS architecture where scheduling and SFs sit within the architecture, showing the process where traffic gets categorized into the SF before being scheduled for transmission. Below is an overview of these SF.

2.9.1 Unsolicited Grant Service (UGS)

Unsolicited Grant Service (UGS) is designed to support real-time SFs aiming to guarantee the bandwidth, latency, jitter and has fixed-sized packets with a constant bitrate. Therefore to meet these real-time requirements the SF incorporates fixed sized grants on a real-time periodic basis. This means that the SS does not need to request bandwidth from the BS, for a specific real-time SF.

The BS can estimate the SF bandwidth demands and pre-allocates the grants according to the QoS parameters negotiated during the connection initialization. Built into the IEEE 802.16 standard are five QoS parameters to define the actual QoS requirements, Maximum sustained traffic rate, maximum latency, tolerated jitter and requested/transmission policy for UGS.

<u>Maximum Sustained Traffic Rate (MSTR):</u> The MSTR parameter holds the peak information rate of the service (IEEE 802.16e-2005, IEEE Std 802.16-2004/Cor1-2005, 2006). This parameter along with the fixed sized grant determines the interval at which the BS issues periodic data grants. The grant size shall be large enough to contain the fixed length data carried by the SF, measured in bps.

<u>Maximum Latency:</u> The maximum latency parameter contains the upper boundary of latency that can be tolerated. Explicitly this is the time it takes for the packet to be sent from the BS and received at the SS. If the time exceeds this boundary then the application will be rendered unusable.

<u>Tolerated Jitter:</u> This is network packet delay variation, which is often referred to as jitter. Here the delay is not constant and the packets arrive in an ad hoc manner. Too much variation in the delay can cause pixilation or frozen frames in TVoIP in the worst case blue screens can be experienced due to data loss. Some jitter can be tolerated in a transmission as the human brain is

resilient to a point before the QoS is deemed to poor. The ITU-R BT1363 standard incorporates jitter requirements for network performance (ITU, 1998).

Request/Transmission Policy: The policy incorporates the rules for the Up Link (UL) bandwidth request and PDU formatting. As bandwidth requests are not required for UGS, the policy should reflect that bandwidth is pre-allocated without request or contention.

Minimum Reserved Traffic Rate (MRTR): Specifies the minimum rate reserved for the SF in bps. This is based on the average minimum traffic rate required for a SF. For example if the MRR is specified as 1.544mbps and the MSTR is defined as 3mbps it is assumed that all packets would be transmitted if the throughput was a constant 1.544mpbs. The transmission can also utilize additional bandwidth until it meets the MSTR at which point further requests for bandwidth will be denied within this example the Minimum Reserve Requirement (MRR) if admitted is guaranteed but the additional bandwidth of MSTR is available on request.

The BS can control the timings of the allocated grants to satisfy the required delay and jitter bound which makes UGS suitable to provide deterministic QoS guarantee for time-critical SF. There is a risk that if the packet generation and unsolicited data grants synchronization is slightly offset from the pre allocated transmission slot, additional delay can be injected into the transmission point of the packet so accurate grant synchronization between BS and SS is important.

2.9.2 Real-time Polling Service (rtPS)

The key aspect of this SF is that it supports variable-sized data packets on a periodic basis. The BS provides Unicast polling opportunities for bandwidth requests which are frequent enough to ensure that latency requirements of real-time services are met. Though this creates higher overheads than UGS it is efficient for variable size data transmission. The rtPS SF also incorporate maximum latency tolerance and maximum sustained rate parameters as detailed in UGS.

2.9.3 Extended real-time Polling Services (ertPS)

This SF was introduced with IEEE 802.16e amendment which provides a periodic UL allocation for particular SS's for data transmission or bandwidth requests. This is not illustrated in the SS diagram in figure 2.11. The SF accommodates data services whose bandwidth requirements may change over time. Similar to rtPS, the ertPS SF also incorporate maximum latency tolerance and maximum sustained rate parameters as detailed in UGS. The difference is that this SF also encompasses jitter tolerance as does UGS, therefore the only difference between UGS and ertPS is the guaranteed bandwidth allocation, making ertPS more efficient for variable data traffic though there is still the issue of additional overhead for transmission grant requests.

2.9.4 Non real-time Polling Service (nrtPS)

This differs from rtPS in that it can also use contention polling in the UL to request bandwidth, including the current queue size. This SF provides minimal bandwidth. Unicast polling is also possible but time between polls is large compared to rtPS (1-2 seconds). As all SS's can request resources during the contention based polling opportunities, collisions can occur. The only QoS parameter that this SF incorporates is maximum sustained traffic rate, as it is assumed nrtPS has limited QoS requirements.

2.9.5 Best Effort (BE)

This SF has minimal QoS support, if any. Data is sent whenever resources are available and not required by any other scheduling-service classes SS uses only contention based polling opportunities to request bandwidth. As for the nrtPS, BE requires limited to no QoS requirements, therefore the only QoS parameter available is MSTR. The constraint of this SF is that it is often starved of bandwidth, resulting in denial of transmission opportunities.

2.10 Dynamic Service Establishment

Incorporated within the WiMAX standard is a signalling function to dynamically allocate SFs. There are three signalling functions; Dynamic Service Activate (DSA); Dynamic Service Change (DSC); and Dynamic Service Delete (DSD). To establish the initial connection the BS sends a DSA-REQ message stating the connection requirements in the form of the SFID, CID and the required QoS parameters (figure 2.10). Applications can utilize the DSC to request changes to the existing resources, including multiple SFs if required to improve the current QoS.

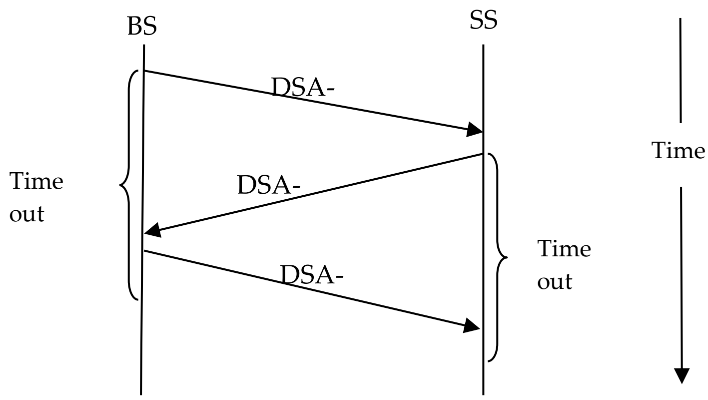

Figure 2.10 Dynamic Service Allocation (DSA)

DSA-REQ and DSA-RSP messages could potentially be retransmitted a multiple number of times if the timeout period expires this is the same for the DSA-ACK message. The aim of deploying these massages is for them to be used without the need to wait for a maintenance window to improve the efficiency in provisioning new services. The retransmission of the dynamic service messages could impact on the transmission latency and potential blocking utilizing the maximum number of retry attempts permitted, (Meucci, Pierucci, & Cerutti, 2010). This, commonly, is the consequence of the interference in the air interface on the radio transmissions or the mobility of the SS resulting in a reduce quality of overall network performance.

2.10.1 Two-Phase Activation Model

The SF activation is designed to work in two phases, 'Admit' and 'Activate'. These are controlled via the Authorisation Module within the BS which accepts or rejects the request. Even if the request is accepted and the authorisation module approves the reservation of the resource required, this does not necessarily allow immediate activation of them but can defer activation.

2.10.2 QoS Parameter Sets

QoS parameter sets that are associated with SFs to distinguish the status granted by the Authorisation module. These include:

<u>ProvisionalQoSParamSet</u>: This relates to QoS parameters that are provided externally to the MAC layer via the network management system, (see figure 2.11).

<u>AdmitedQoSParamSet</u>: This is when the resources that have been requested by the BS and the SS and that are previously reserved (provisioned) for use,

are admitted to the connection but not yet activated. Primarily this parameter set is a subset of the ProvisionalQoSParamSet, as not all requests will be admitted, (see figure 2.11).

ActiveQoSParamSet: This parameter signals the resources that are actually being provided to the SFs. The ActiveQoSParamSet is a subset of the AdmittedQoSParamSet as some of the Admitted resources will defer activation, (see figure 2.11).

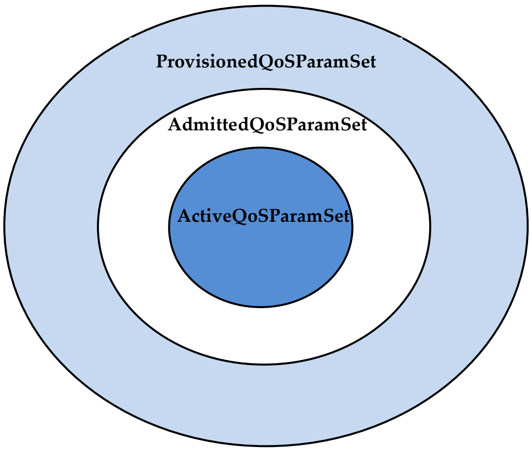

Figure 2.11 The relationship of QoSParamSet of resource allocation per variable

The variables that are incorporated within the QoSParamSet are latency, jitter, and throughput guarantees. The application requirements determine the individual parameters of each variable that in turn determines the SF.

2.10.3 Authorization Module

The authorization module determines which QoS parameters are allowed. There are two types of authorization methods, as follows.

Provisioned Authorization: QoS parameters are provided by the network management system upon setup and remain static.

Dynamic Authorization: Changes to QoS parameters can be requested but it is the authorization module that determines whether the request can be granted.

These mechanisms offer flexibility into QoS provisioning, by providing a three level QoS parameter set. Firstly by checking there is the resource provision available for the QoS request, secondly admitting the resource prior to activation, finally providing a dynamic resource allocation process. The constraint of this model is that resources can be reserved but not used. This is only a problem if the system becomes saturated and the reserved resources cannot be redirected to a SS that initates a connection request that is rejected, as redundant resources were reserved for another SF but not currently being use, which will compromise the QoS of the network. The result is that available resources are underutilized and the network is working under capacity.

The actual QoS requirements are determined by the higher application layer e.g. VoIP will request real-time SF with fixed-sized data grants, FTP would request non real-time SF with variable sized data grants. The requirements can be requested via the parameter set or a service class name. Once the request has been authorized the provisioning is then apportioned via the MAC layer.

2.11 IEEE 802.16 MAC Layer

The MAC layer assigns the traffic to the SF which is then mapped to connections, in order to transform connectionless IP and UDP protocols to behave as connection oriented protocol. Therefore the IEEE 802.16 utilizes a connection oriented protocol where services flows can represent an individual application or groups of applications with the same CID, (Wongthavarawat &

Ganz, 2003). To accomplish this, the Mac layer is divided into two sub-layers the convergence sub-layer and the Common Part sub-layer (figure 2.14). The MAC layer is compatible with both ATM as well as IP packet service incorporating both IPv4 and IPv6. The compatibility with ATM is significant as it incorporates switching techniques that are used in the telecommunication networks. Encompassing functionality of both circuit switched networking and packet switched networking technologies.

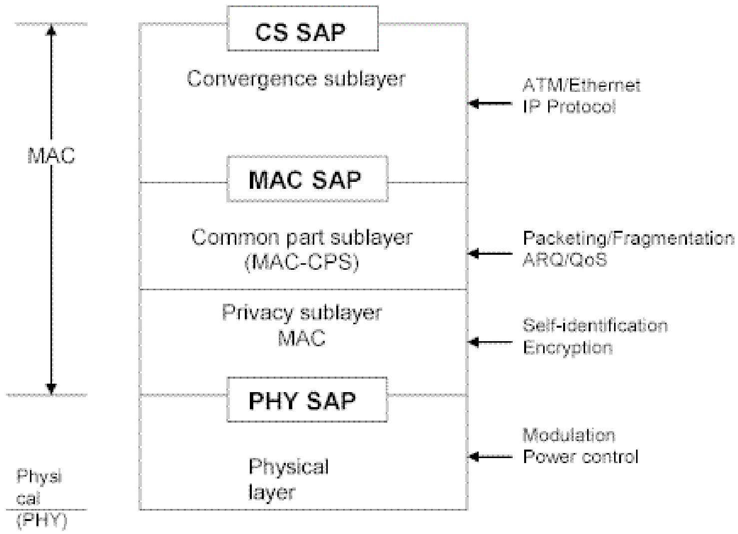

Figure 2.12 IEEE 802.16 Implementation of the Reference Model with WiMAX

2.11.1 Convergence Sublayer

The aim of this sub layer is to map the transport layer into the core common part sublayer which supports the convergence of ATM cell and IP packets. The service data unit (SDU) is classified by the traffic type before being allocated a SF which is identified by a 32 bit SFID. When the QoS requirements of the SFs are either admitted or activated, they are then mapped to connections that can provide the QoS requirements each connection is given a CID. Once the CID is assigned to a SF it is then forwarded to an appropriate queue (figure 2.14)

This sublayer is independent of the transport mechanism and manages the fragmentation and segmentation of the SDU’s into the MAC protocol data unit’s(PDU), QoS control, scheduling and retransmission of MAC PDU’s (Ghosh, Wolter, Andrew, & Chen, 2005). The PDU’s can be concatenated into bursts with the same modulation and coding. The QoS control covers the bandwidth requests and grant mechanisms.

2.12 Scheduling

Uplink scheduling is to managed by the BS then assigning it to the SS. The BS will then indicate the time slots for the SS to transmit via a UL MAP massage. The SS then retrieves the packets from the queues, and then synchronizes the transmission of the packets to the allocated time slots provided by the BS. A mechanism to improve packet latency by reducing the transmission of redundant data is done by the suppression of the packet header data. This is only possible once the CID is assigned to the SF to determine the transmission route.

2.12.1Bandwidth Requests

The SS will request bandwidth for each connection using a bandwidth request message. This can be sent as a standard request or a piggybacked packet. Each request can be up to the value of 32kbs. If the SF is UGS it does not require a bandwidth request as the QoS requirements are implicit when the connection is activated so no requests are used. The other SF’s have different bandwidth request mechanisms, rtPS provides periodic bandwidth request opportunities where as nrtPS incorporate a more random approach to transmitting bandwidth request opportunities. Each mechanism reflects the QoS transmission needs of each SF requirement. The request can be either incremental by asking for additional bandwidth or aggregated where the bandwidth requirement of the connection are collected into one request. The aggregated requests can provide the mechanism for the BS to efficiently correct its perception of the SS's actual requirements. Within the bandwidth request header the SS can ask the BS to poll it. The BS is able to poll the SS individually or as a group.

The polling methods utilized for transmitting bandwidth requests are: unicast, multicast, broadcast and station initiated polls. Unicast enables the BS to check for inactive SS's, multicast or broadcast is used for efficiency when polling individual SS's with constrained bandwidth resources, therefore, are polled collectively. The SS initiated poll is instigated by a poll-me-bit to request the BS to poll the individual SS, developed primarily for UGS SF but the SS often uses this for other SF, that has an individual mechanism to deliver its bandwidth request table 2.7 illustrates these methods

UGS	BS provides fixed sized data grant bursts periodically
rtPS	BS provides the SS opportunity to request bandwidth on a regular basis
nrtPS	BS provides SS opportunity to request bandwidth using unicast and contention methods
BE	BS allows SS to use all available mechanisms for transmission request unused by other SFs.

Table 2.7 Bandwidth request mechanisms per services flow type.

2.12.2 Bandwidth Grants

Two methods of bandwidth grants are defined by the IEEE 802.16 standards Grants Per Connection (GPC) and Grants Per SS (GPSS). The GPC provides a method for the BS to grant bandwidth exclusively for a connection while

GPSS provides a mechanism for the BS to grant aggregated requests to the SS. With an aggregated request, the SS manages the distribution of the bandwidth among its connection to maintain the required QoS.

The GPC mode does provide for a simpler system that focuses on one connection thus reducing the computation required at the SS with bandwidth redistribution. This mode works well when the numbers of users are relatively low. The disadvantage of this techique is it has higher overheads than GPSS (Freeman, 2005). GPSS can cope with many more connections per SS. As the SS has the responsibility of redistributing the bandwidth, this makes the mode scalable and more reactive to the application QoS requirements. The overheads for this mode are lower but it requires an intelligent SS which is capable of redistributing the bandwidth. Both modes apply a self-correcting protocol, for instance there is no waiting for acknowledgement, therefore after a timeout period the request is retransmitted. This eliminates the overhead of the acknowledgement as utilized in TCP, thereby reducing the bandwidth usage.

Grants are given as durations, carried on UL MAP messages sent from Uplink (UL) packet scheduling systems of the BS to the packet scheduler of the SS. The grants are provided in the form of TXOP in the UL-MAP. The provision of the bandwidth is determined by initially evaluating if the availability of the network resources can grant the amount of bandwidth requested to meet the requirements of the SF.

2.13 Scheduling Algorithms

Scheduling in a wireless environment has many challenges due to the air interface that the transmission has to pass through. The vulnerabilities inherent in the wireless domain make efficient use of radio resources a challenge. The fundamental factor which determines network performance relates to scheduling algorithms. It was inferred from Grigorik's (2011) data (see figure 2.1) that the bandwidth availability alone does not provide a good QoS. In fact, it is the efficiency of the algorithms coupled with the required bandwidth that enhances the performances (Grigorik, 2012). Therefore, the aim of a scheduler is to control the allocation of the resources to attempt to acquire the diverse QoS requirements while accommodating the variability of the wireless channel conditions (Fattah & Leung, 2002). As illustrated in figure 3.1 the Up Link (UL) scheduling for rtPS, nrtPS and BE is still undefined within the standard including ertPS. To achieve a performance that will cope with the quality demands of multimedia traffic efficient algorithms need to be utilized by the BS. The BS performs scheduling on a per connection basis; therefore the majority of the algorithm requires some key data to include the number of total connections; the number of pending connections; per connection throughput requirements and the connection queue status. This data is ascertained when the SS connects to the BS. Classical scheduling algorithms such as Weighted Round Robin (WRR) (section 2.2.3) and Weight Fair Queuing (WFQ) (section 2.2.4) are ideal, in providing QoS, but the scheduling uplink flows are more complex due to the location of the queues as it requires the use of the UL-MAP to distribute the scheduling requirements.

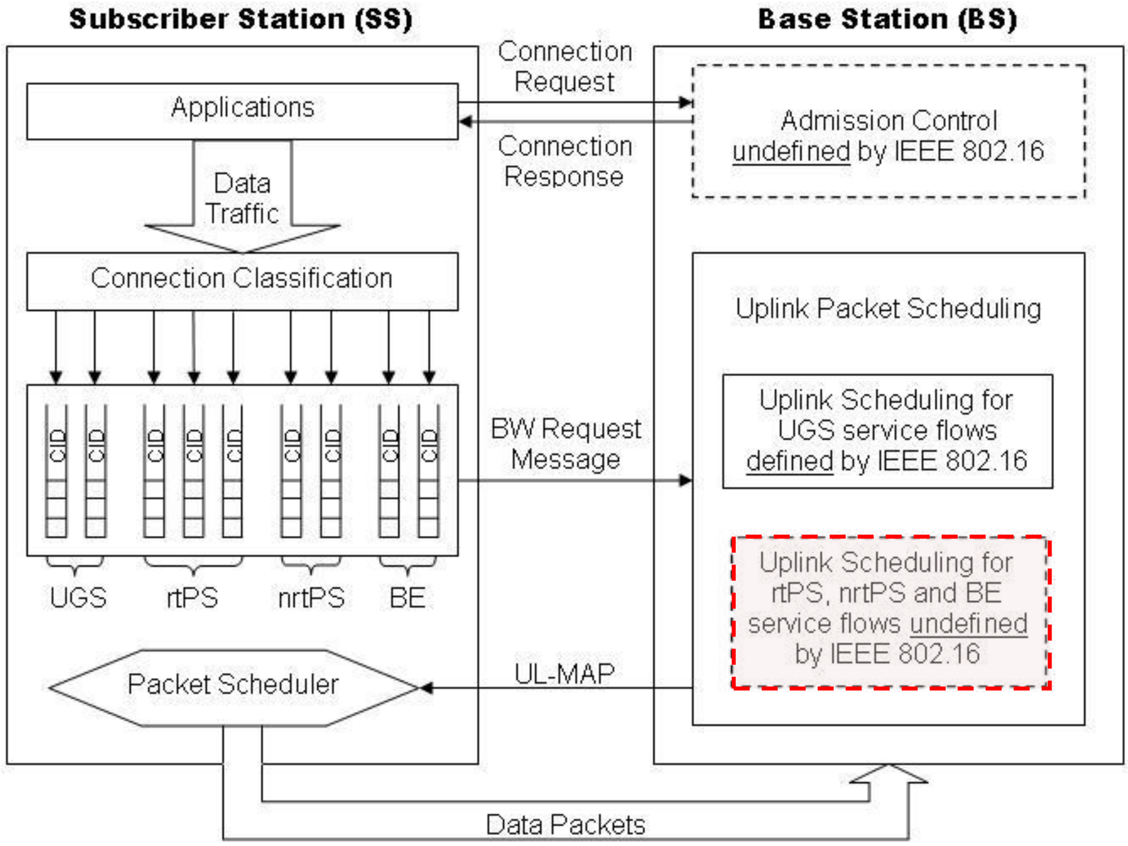

Figure 2.13 IEEE 802.16 QoS Architecture (IEEE 802.16e-2005, IEEE Std 802.16-2004/Cor1-2005, 2006)

To achieve an optimal system performance with an appropriate amount of fairness throughout the PHY and MAC layers. Cao and Li (2001) identifies five key issues in wireless networking, these are: the variability of the actual wireless link; the fairness of the throughput of the transmission; the QoS to guarantee the required parameters are available for the transmission whatever the condition of the network; and the data throughput and channel utilization ensuring efficient use of the available network resources and power capacity (Cao & Li, 2001).

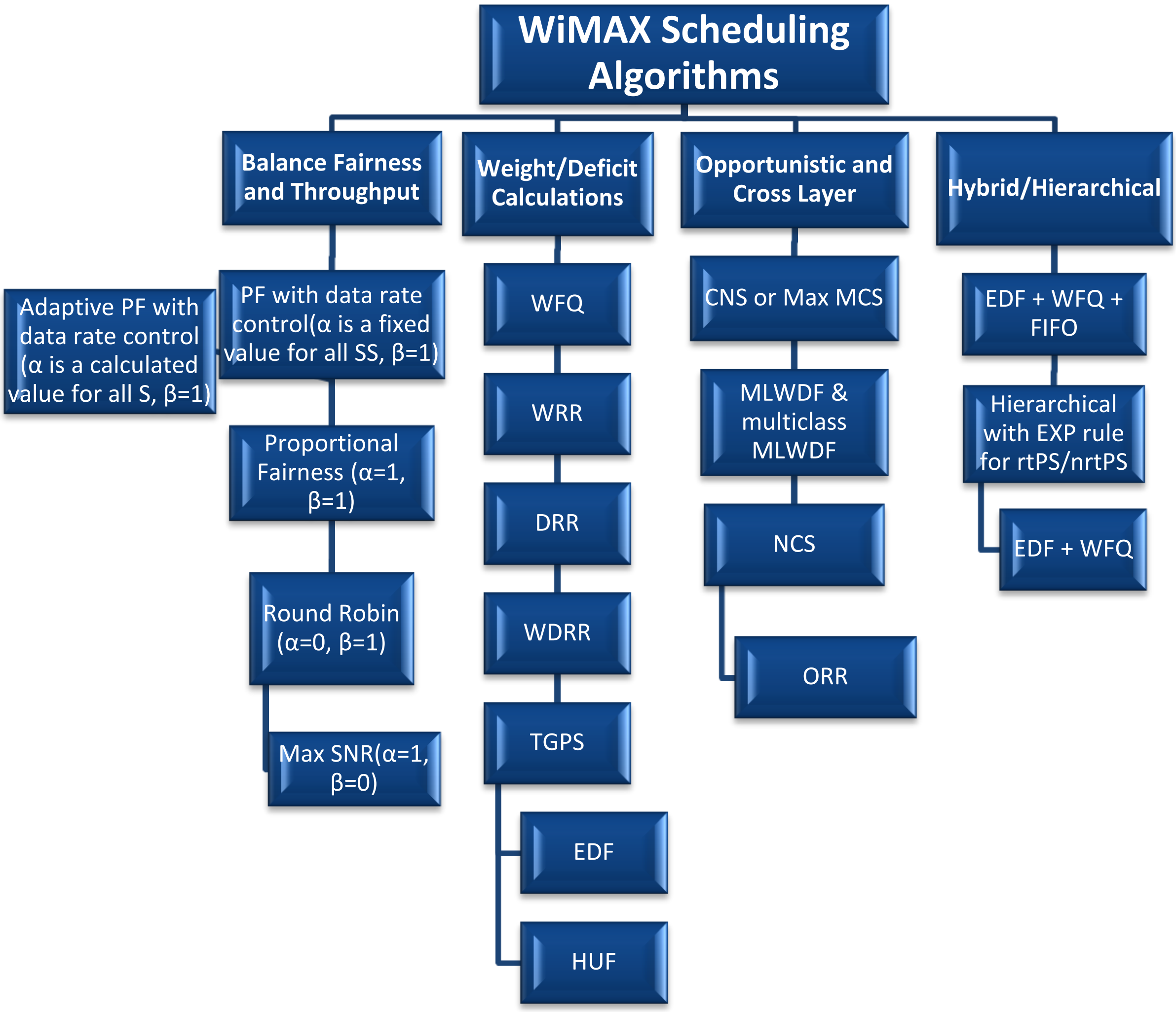

Figure 2.14 Taxonomy of Scheduling Algorithms for WiMAX adapted from (Dhrona, Abu Ali, & Hassanein, 2008)

Figure 2.14 illustrates a taxonomy of scheduling algorithms which is adapted from Dhrona et al (2008) performance study of scheduling algorithms in WiMAX. These categories have been selected based on their similarities of the method and parameters used for decisions. The four categories defined are Balance Fairness and throughput, Weight/Deficit calculations, Opportunistic and cross-layer and Hierarchical and hybrid (Dhrona, Abu Ali, & Hassanein, 2008). Though the initial algorithms are simpler in their application this making them easier to categorize the modern algorithms which are more complex, making them more difficult to categorize. Other taxonomies have been defined in the literature, Jain, So-in and Tamimi (2009), developed a taxonomy based on channel information defining two categories: (i) Channel Aware which is sub-divided into fairness, QoS guarantee requirements, maximising throughput and efficient power utilizing and (ii) Channel Unaware (Jain, So-in, & Tamimi, 2009). The drawback of this categorization is that many of the scheduling algorithms can easily fit into both categories, for instance WFQ can also be deployed with an adaptation which would calculate the dynamic channel weightings. With the adaptation WFQ would then fit into the channel aware category where a traditional implementation of the algorithm would mean it would be a Channel Unaware. To avoid such ambiguity a modification of Dhrona et al (2009) taxonomy was adapted to aid its clarity by basing it on the method of operation figure 3.2 (Dhrona, Abu Ali, & Hassanein, 2008).

2.13.1 Balanced Fairness and Throughput Algorithms

The criteria of this category is to ensure all SFs and channel connections maintain their throughput without any of the lower priority SFs such as BE and nrtPS being starved of resources. The fairness and throughput algorithms focus on an efficient balance between fairness and throughput. Fairness is determined by the resource availability and channel conditions. Therefore fairness is conditional and not absolute (Anderlind & Jens, 1997), as an explicit fair distribution of resources can lead to inefficiencies.

2.13.1.1Round Robin

This is a conventional cyclic algorithm, (Nagle, 1987), where users are allocated an equal time slot of a predetermined quantity in an organised ring, (see figure 2.15). Each user’s allocated resources has to be consumed before additional slots are granted. This promotes ‘fairness’ as it provides equal resources distribution. The weakness part of this algorithm is that the resources can be allocated to the users with poor channel conditions which results in inefficient allocation of resources. This is due to the fact that poor channel conditions will negate the efficiency of the resources allocated. Within the WiMAX domain this scheduling algorithm would therefore ignore the priorities of SF, meaning the BE SF would be treated equal to the rtPS SF. This would ensure that BE will not be starved of resources but in turn would inject inefficiency into the rtPS SF, for example if a video was being streamed and the connection used up its allocated time slots the remainder of the video would incur latency until its queue allocates additional time slots to continue the video stream, this scenario though would treat all SF equally, for the higher priority SF QoS is reduced.

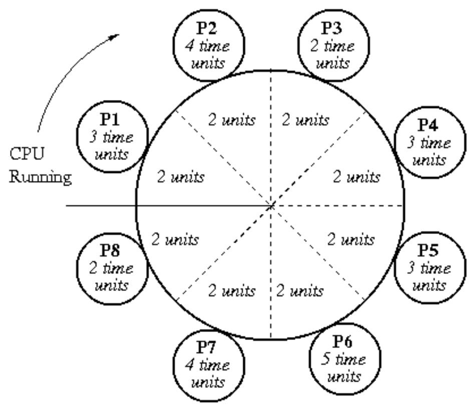

Figure 2.15 Round Robin scheduling example (Kollewin Blog, 2009)

2.13.1.2 Maximum Signal Noise Ratio (MaxSNR)

This algorithm ranks users in terms of their Signal Noise Ratio (SNR). When scheduling the user with the highest SNR value is provided with timeslots depending on QoS requirements and resources availability. The disadvantage is that the same user can be granted successive slots if they qualify with the highest SNR value. Conversely this algorithm does provide high system throughput (Hamdi, 2007). Though this algorithm does consider QoS requirements, in the form of channel conditions, it is susceptible to starving high priority SF if they do not have high SNR. Therefore if a low priority SF, such as nrtPS, with high SNR would qualify for successive slots starving the higher priority SF.

2.13.1.3 Proportional Fair Algorithms

The aim of proportional fair algorithms is to take the weaknesses of the above algorithms and attempt to balance the efficiency of the throughput by incorporating conditional fairness based on a number of factors, as illustrated by the equation below (Jalali, Padovani, & Pankaj, 2002), where the current data rates supported by the channel are averaged out to promote fairness.

$$j = \frac{\arg \quad max}{1 \leq i \leq N} \frac{D_i(t)^{\propto}}{R_i(t-1)^{\beta}} \qquad (2.2)$$

For each user (j) the algorithm (2.2) determines a proportional allocation of resources based on the total number of users (N), (assuming 1 ≤ N). The distribution is determined by the current data rate that the channel supports, $D_i(t)^{\propto}$, where ($\propto$) incorporates the traffic shaping of the channel by limiting the data rate. Whereas $R_i(t-1)^{\beta}$ is the actual data rate per unit of time (t) for each instance of the user (i). In this case (β) is utilized to control the data rate as the averaging parameter over the unit of time.

The algorithm decides the selection of the user by establishing if it has good channel conditions determined by $D_i(t)^{\alpha}$. Users with low average data rates $R_i(t-1)^{\beta}$ will increase their chance of being selected for the next scheduling event. Both Holtzman (2001) and Rentel et al (2002) research reflect the issue of effective fairness in this algorithm is still not resolved (Holtzman, 2001), (Rentel, Krzymien, Darian, Vanghi, & Elliott, 2002). Though the users with the lower data rates have a better chance of being selected the above issue with fairness remains with additional latency will be injected into the transmission compromising QoS. Kuenyoung et al (2002) method to resolve the problem involves the use of adding an exponent to the Data Rate Control (DCR) element modifying the algorithm, where (n) becomes a weighted parameter introduced to balance the data rate with the channel conditions, (2.3) (Kuenyoung, Hoon, & Youngnam, 2002).

$$j = \frac{\arg \quad max}{1 \leq i \leq N} \frac{D_i^n(t)^\propto}{R_i(t-1)^\beta} \quad (2.3)$$

The drawback of this algorithm results from the fact, (n) is fixed in time and not adaptable to the current channel conditions. In the attempt to address the fairness issue (n) being a constant, produces an inefficient allocation of resources but to a lesser extent than equation (2.2). Zhu and Hafez (2006) concluded that fairness is not guaranteed all cases but only in specific ones (Zhu & Hafez, 2006).

The algorithms reviewed so far do not attempt to incorporate prioritization of scheduling QoS services. Though the channel conditions are a major factor in the transmission it is not the only consideration. When transmitting, streaming media requirements needs to be supported, therefore the inefficiencies identifed are still prevalent in this scheduling algorithm, even though it aims to incorporate fairness. It is apparent from this algorithm that fairness cannot be determined on a single key parameter.

2.13.1.4 Adaptive Proportional Fairness Algorithms

The aim of the adaptive proportional fairness algorithm is to track the data rates allocated to each user and update the exponent parameters in order to achieve fairness among the users based on their QoS requirements (Aniba & Aıssa, 2004), whereas the traditional proportional fairness algorithm does not consider QoS in the selection criteria. It utilizes BE and maximizes the total data rates while providing fairness among the community user. The adaptive proportional fairness algorithm uses a control parameter which replaces Kuenyoung's et al (2002) exponent parameter for each user in order to avoid the dependency between the different users. Together with this a monitoring module is included to track the variable conditions of the communication channels, encompassing the different users (Kuenyoung, Hoon, & Youngnam, 2002).

$$j = \frac{argmax}{1 \le i \le N} \frac{D_i^{c_i}(t)^{\propto}}{R_i(t-1)^{\beta}} * RT_i \quad (2.4)$$

The equation (2.4) provides a monitoring module which updates the Value C_i, $i = 1,\ldots N$ to account for the equality between the proportional allocated data rate values. RT_i denotes the target data rate of each user, this considers limited QoS requirements within the selection process. By enhancing this element of the algorithm, Aniba and AÏssa (2004) demonstrated a higher efficiency of fairness while not degrading the utilization of the throughput (Aniba & Aıssa, 2004). Variability of bandwidth is considered so each connection will be given the requested resource but the SF prioritization is still not a consideration within this algorithm therefore BE can be served before rtPS. Therefore scheduling within the WiMAX paradigms incorporate a parameter that will test the priority of the SF as part of the algorithm but in turn this would contradict the rationale of fairness.

2.13.2 Weighted/Deficit Calculation based Scheduling Algorithms

The proportional fairness algorithms have been based on determining the bandwidth requirements and the SS current bandwidth capacity for decision making in scheduling. Zhu and Hafel (2006) identified the shortcomings of this algorithm, while Kuenyoung (2002) proposed an adaption to improve the proportional fairness algorithm, by introducing the control parameter and monitoring module but this was still based on the best effort paradigm, while still not accommodating the QoS prioritization requirements of the transmission (Kuenyoung, Hoon, & Youngnam, 2002). Therefore, to further extend the proportional fairness algorithm Weighted/Deficit calculations utilize algorithms based on Generalized Processor Shaping (GPS), where the decisions are determined by the weights, delay or deficit measurements that are allocated dynamically considering different QoS requirements.

2.13.2.1 Generalized Processor Shaping (GPS)

GPS was developed as a service discipline to share the capacity of the congested communication links in an efficient, flexible and fair manner (Parekh & Gallager, 1993). The basic principle is that a fixed weight is assigned to each user, instead of the fixed bandwidth and to allocate bandwidth to all the users according to their individual weights and traffic load (Parekh & Gallager, 1993). The need for quality transmission is addressed by this weighted variable, which can be measured by service priorities, fairness considerations, bandwidth and delay requirements. The calculation of these weighted variables as the decision making part of the scheduling algorithm provides flexibility to deliver better QoS compared to Proportional Fairness algorithms. The GPS algorithm is utilized as the benchmark against which reliable fair scheduling disciplines can be measured, such as Weight Round Robin (WRR) (Demers, Keshaw, & Shenker, 1989). The constraint on the algorithm is, that it relies on fluid traffic. The users can be aligned with a particular connection requirement but the constraint makes this algorithm impractical for a connection oriented paradigm with no requirements for a continued flow of traffic. As the user has the fixed weighting the SF priorities would also be overlooked within this scheduling algorithm.

2.13.2.2 Truncated Generalized Processor Sharing (TGPS)

Cai, Shen and Mark (2003) proposed the Truncated Generalized Processor Sharing (TGPS) which incorporates circuit-switched OFDM by supporting single user's with multiple QoS traffic requirements (Cai, Shen, & Mark, 2003). This took the GPS algorithm, integrated power allocation and subcarrier allocation while still providing fairness to all traffic, incorporating maximum throughput whilst guaranteeing SNR for heterogeneous traffic and satisfying the total transmission power constraints. The GPS scheduling algorithm assumes that all users can be served simultaneously in small increments. With OFDM and parallel transmission it is possible to achieve this using GPS, but there are two constraints to its successful deployment. Firstly there is a limitation to the maximum number of users that are transmitted simultaneously. The algorithm divides the total bandwidth into a number of subcarriers M. Due to server channel fading the number of subcarriers supported is limited to less than M, as the total transmission power is limited. If the backlog of users is greater than those that actually can be accommodated by available bandwidth, some users will be denied resources that should be guaranteed under the GPS algorithm. Secondly, even if the total number of backlogged users is less than what can be accommodated by the subcarriers, the scheduling algorithm may still not be able to guarantee the capacity allocation according to the predefined weights, especially if the required capacity does not result in an integer number of subcarriers.

$$M_i^k = \frac{\emptyset_\mathrm{i}}{\sum_{\mathrm{j} \in \mathrm{B}} \emptyset_\mathrm{i}} \mathrm{M}_{\mathrm{eff}}^{\mathrm{k}} \qquad (2.5)$$

The TGPS algorithm incorporates the constraints of the GPS algorithm as shown by equation (2.5) (Cai, Shen, & Mark, 2003). Here, M_i^k provides the allocation of resources for user i at time k, while M_{eff}^{k} denotes the number of effective subcarriers available, as a consequence of the channel fading, of time k, in a busy time period. $Ø_i$ is the weight assigned to the user i, which is also weighted against all active users represented by $\sum_{j \in B} Ø_j$. Therefore the service ratio for user i is determined in proportion to its weight and the aggregated weight of other active transmissions. This equations (2.5) accommodates the capacity allocation as determine in GPS, but there is also a loss of capacity, from each backlogged user which is indicated by $\{\mathcal{E}_i^k, i \in B\}$. $\mathcal{E}_i^k$ is derived from the quantization of the OFDM system within the frequency domain and therefore termed as the quantization error. The accumulated quantization error reflects the total difference between the TGPS and GPS scheduling algorithms, see equation (2.6) (Cai, Shen, & Mark, 2003). Here a compensation algorithm is initiated to allocate the remaining subcarriers until the quantization error equals zero. It is also set to zero, once a user completes the transmission of all its packets as the error value has been translated into the transmission delay of the packets, so it is not considered for future transmission allocations.

$$\mathcal{E}_i^k = \frac{\emptyset_\mathrm{i}}{\sum_{\mathrm{j} \in \mathrm{B}} \emptyset_\mathrm{i}} \mathrm{M}_{\mathrm{eff}}^{\mathrm{k}} - M_i^k \qquad (2.6)$$

Therefore TGPS can guarantee the capacity allocation to all the backlogged users at any time instance of k according to the predefined weight $Ø_i$ and provide fairness among users. The restriction of the number of user denied a connection conflicts with the aim of the proposed harvesting redistribution algorithm (see Chapter 6) which aims to reduce the number of connections denials.

2.13.2.3 Weighted Round Robin (WRR)

WRR (Katevenis, Sidiropoulos, & Courcoubetis, 1991), is an abstraction of the RR algorithm, it is also an approximation of GPS, and therefore has similarities to TGPS. Each connection has its own weighted packet queue, where the algorithm distributes the bandwidth between the queues based on the allocated weight associated to it representing its priority. Though the goal is to use a weight for each subcarrier with packets queued to transmit, it does not allocate the actual subcarriers. The weighting is determined on the MRTR of the subcarrier (2.7), the algorithm iterates giving connections to the SS with the higher weighting first.

$$W_i = \frac{MRTR_i}{\sum_{j \in n} \mathrm{MRTR_j}} \qquad (2.7)$$

Here, W_i is the weight assigned to user i and the total number of subscribers. In the equation $MRTR$ this is the minimum rate that the SF requires to send its packets appropriately. The weighting in this algorithm does not consider QoS parameters required to determine SF priority requirements.

2.13.2.4 Weighted Fair Queuing (WFQ)

Weighted Fair Queuing (WFQ) is also an abstraction of the RR scheduling algorithm and is used within the QoS architectures (Parekh & Gallager, 1993). The algorithm is based on the bandwidth requested by each connection. The packets are classified and queued in the relevant class queue. The classes are then served in a circular manner, if a queue is empty this is then skipped and the next queued packet will be served. WFQ can provide a differential service per class, for example UGS is served first, and then other classes are allocated their bandwidth based on their weighting. Each connection i is assigned a weight W_i ,,and BW_i is the requested bandwidth by each connection and n is the total number of connections.

$$W_i = \frac{BW_i}{\sum_{j=1}^{n} BW_j} \qquad (2.8)$$

Where, $\sum_{j=1}^{n} BW_i$ accounts for the total number of connections and bandwidth requests. The total bandwidth is shared between each connection which will receive a fraction of the service available during any time interval. This algorithm ensures that even in low bandwidth availability the queues will receive a share of the bandwidth. This algorithm was incorporated utilising OFDM in WiMAX domain but its complexity was deemed too high. Also it had been applied to OFDMA where several connections can be served simultaneously during a single frame. The drawback is that it would require multiple iterations of the algorithm, this further increased the complexity of the algorithm in this scenario (Dhrona, Abu Ali, & Hassanein, 2008). WFQ does not account for UGS, because this is served first and the rest of the bandwidth can be distributed between other SF requirements. Though a weighting is allocated based on the bandwidth requested per connection, where the constraint of the algorithm is that it does not reflect the priorities of the service classes within in WiMAX, making it difficult to prioritizes between rtPS, nrtPS and BE, which needs differing SF.

2.13.2.5 Deficit Round Robin (DRR)

The Deficit Round Robin scheduling algorithm was originally developed for IP networks, (Shreedhar & Varghese, 1996). This algorithm addresses those connections that have not been scheduled and therefore should accommodate the SF starvation of resources that is apparent in other algorithms. This is accomplished by increasing a deficit counter, which utilizes a unit of measurement referred to as a quantum unit. The deficit counter is used to compare the size of the Head of the Queue (HoQ) of each active connection. If the deficit counter is greater than the HoQ packet, it will be scheduled and the deficit counter is decreased by the size of the packet, the packets are continued to be sent as long as the deficit counter remains larger than the HoQ. If all the packets are sent for that connection the deficit counter is then set to zero. Otherwise the queue will be skipped and the deficit counter is incremented by a quantum unit. The counter and HoQ will be compared again on subsequent iterations until it is able to be scheduled. Compared with the fair queuing scheduler which has a complexity of $O(\log(n))$, (n being the number of active flows), the complexity of DRR is $O(1)$ (Shreedhar & Varghese, 1996). DRR operates with varying packet sizes but relies on the knowledge of HoQ per connection which needs to be elicited from the SS, for this reason the DRR needs modification to be utilised within the WiMAX domain.

2.13.2.6 Weighted Deficit Round Robin (WDRR)

Weighted Deficit Round Robin (WDRR) is a variation of DRR, including the quantum size is adjusted according to the connection Modulation and Coding Scheme (MCS). This is accomplished by multiplying the quantum by bytes per slot that the current MCS of the connection can deliver and then divide the quantum by six. Preference is given to those connections with higher quantum count than connections with a lower MCS for scheduling. The limitation of this algorithm is that it is prone to connection starvation (Lakkakorpi, Sayenko, & Moilanen, 2008). WDRR manages queuing and is powerful enough to accommodate the granular needs of QoS, but the QoS complexities of incorporating this into the WiMAX domain remains.

2.13.2.7 Earliest Deadline First (EDF)

Earliest Deadline First (EDF) is a dynamic scheduling algorithm which is the most widely used scheduling algorithm for real-time applications as the selection is based on the SS delay requirements, (Nie, Xiong, & Wang, 2010). EDF assigns deadlines to packets on each connection and allocates the bandwidth to the SS based on these deadlines. EDF is sensitive to delay as the priority of the scheduling increases with the time period it spends in the queue. This algorithm is more applicable to UGS and rtPS as these classes have stringent delay requirements. The disadvantage is if the system is overloaded the packets that will miss the deadline is unpredictable, but the algorithm can distribute the overload more fairly. One key issue is the shortest deadline might not be the most important or the highest priority packets to send first to maintain QoS. For this algorithm to be effective in WiMAX the SF priorities would need to be considered alongside the delay constraints.

2.13.2.8 Highest Urgency First (HUF)

The Highest Urgency First (HUF) is a modulation, latency and priority-aware algorithm incorporating an 'Urgency' parameter. This promotes the idea that latency-dependant flows do not necessarily need to be served first. It is paramount that these flows still stay within their delay tolerance period. This algorithm utilizes bandwidth allocation, encompassing the higher capacity of OFDMA's flexibility in the mobile environment. The algorithm has four key stages (Lin, Lin, & Lai, 2009): Firstly, bytes are translated into slots, reflecting the MCS of every SS, while also calculating the number of frames to ensure it complies with the maximum latency requirements for the SF. Secondly, calculating and proportionally allocating the number of slots required by DL/UL requests for transmission on the DL/UL subframe. Thirdly, the algorithm allocates slots for the SF using the U-Factor (urgency parameter) which indicates the latency, priority and fairness of all bandwidth requests. Lastly, the algorithm allocates the slots of flows to the SS. These key stages of the algorithm are then encompassing two phases of execution. The first phase determines the bandwidth of the DL/UL subframe. The second allocates bandwidth for requests from the SS each phase utilizes different metrics for decision making.

The U-factor uses three metrics to decide the order of the requests received, deadline (latency, a deadline with a value of one must be transmitted on the current frame), fairness (the number of slots) and priority (priority of SF). The deadline is established from the number of frame durations left before the UL/DL requests are served. The priority is measured by the size of the data to be sent. The assumption is that requests of large bandwidth will be served first as it is easier to accommodate smaller bandwidth requests. The disadvantage of this assumption is that there is a high risk of the smaller bandwidth requests missing their deadline all together. It also suffers from degradation of throughput and fairness. As this algorithm is focused at scheduling from the BS perspective it also requires an accompanying SS algorithm to schedule the appropriate granted bandwidth.

2.13.3 Opportunistic and Cross-Layer Algorithm

Opportunistic algorithms aims to distribute resources in a wireless network that take advantage of instantaneous channel variations by giving priority to the user with favourable channel conditions (Fattah & Leung, 2002), (Hassel, 2007).

The challenge within the wireless domain is the time varying channel conditions and multiuser diversity. The wireless channel was originally modelled by two states using Markov Chain (Gilbert, 1960), a user experiences error-free transmission when it observes a ‘good’ channel, and an unsuccessful channel categorized as a ‘bad’ channel. In today’s complex wireless environment this model is too simplistic. Opportunistic scheduling utilizes the fluctuations of channel conditions, therefore scheduling gain depends on the amplitude of the channel conditions, therefore the greater the number of users the better the performance (Lu, Ma, & Gong, 2009). The four prominent opportunistic algorithms all utilize signal quality as their key parameter input (8) (Hassel, 2007);

$$i * (t_k) = \frac{arg\ max}{1 \le i \le Nai} \frac{X_i\ (t_k)}{a_i} \quad (2.9)$$

Equation (2.9), denotes $X_i\ (t_k)$ as the metric function, this is calculated at the beginning of the time slot $(t)_{k,}$. Where $i^*(t_k)$ indicates the user selected for scheduling, with a_i being the constant of the channel (Hassel V. , 2007). The fluctuations in the channel conditions needs to be stable enough for the user to estimate it, if the channel conditions fluctuates rapidly it is anticipated this would mean that latency would not be increased significantly, due to the trade-off between scheduling gain and short term performance. The stronger the stability of channel conditions (slow channel fluctuation) and the worse the short term performance, high fluctuation of channel conditions only produce greater improvements in short term performance but less in scheduling gain. The cost of the scheduling is the consequence of reporting the state of the channel conditions constantly, which needs to be balanced with the scheduling gain. Also estimation errors can occurs these include channel, parameter estimation errors as well as errors due to transmission delay. Consequently the accuracy of estimation is better if the channel conditions are more stable.

2.13.3.1 Carrier to Noise Ratio (CNR)

Carrier to Noise Ratio (CNR) is where the signal strength of the received signal is relative to the noise from the bandwidth (Goldsmith, 2005) this helps to ascertain the quality of the transmission channel. Here C/N is equivalent to S/N based on Shannon's theorem. In the equation (2.10) below P_r indicates the received power in the form of decibels per milliwatt (dBm). Whereas, N_0 denotes the noise power spectral density in the form of *dBm/Hz*. Whereas W is the received signal bandwidth and P_1 expresses the sum of the received power associated with the intercell and intracell interference (Goldsmith, 2005).

$$\gamma = \frac{P_r}{N_{0W+P_1}} \qquad (2.10)$$

The BS selects the user with the largest CNR value of the all the users. If the CNR of the user falls in the rth fading region, the constellation M_r is transmitted. The region boundaries, equation (2.11) are set to the CNR ensuring the Channel State Information bits are set below the Bit Error Rate (BER).

$$\text{Region boundaries} = \{\gamma_r\}\begin{matrix} R+1 \\ r+1 \end{matrix} \qquad (2.11)$$

If the CNR is high it will provide the benefit of quality of signal delivery and accuracy of the actual communication, together with good reliability, than channels with low CNR. CNR is a cross layer approach that uses the information on the channel quality from PHY layer this is calculated as the channel specification (Cspec) and the traffic specification (Tspec) to ascertain quality scheduling (Lu, Ma, & Gong, 2009). However the Tspec is used for scheduling those connections with a poor CNR which could still be starved of TXOP, Therefore, a high priority SF with poor CNR could possibly breach their latency requiremnts with this scheduling algorithm.

$$i(t) = \frac{\arg\max R_i\,(t)}{1 \le i \le N} \qquad (2.12)$$

2.13.3.2 Maximum Carrier to Noise Ratio (Max CNR)

The Maximum Carrier to Noise Ratio (Max CNR) schedules the user with the highest CNR in each time slot (Knopp & Humblet, 1995), thus will providing similar benefits to CNR but negate this priorities of the SF when selecting the user. This algorithm takes advantage of the maximum average system spectral efficiency algorithm which is the theoretical throughput of each bandwidth connection averaged over the total number of users in the system (Goldsmith & Varaiya, 1997). Within the Max Constant Bit Rate algorithm the MCS policy takes advantage of the multiuser diversity in a time slot if one user is scheduled at a time. The parameter input for this algorithm is represented in equation (2.9). This algorithm upholds fairness only if the carrier to noise ratio is i.i.d, again those with lower CNR will not get scheduled which in turn reduces QoS.

2.13.3.3 Normalised Carrier to Noise Scheduling (NCS)

As CNR and Max CNR aims to schedule the highest ratio this does not provide fairness to scheduling, since users could also be starved of resources to transmit. Therefore the Normalised Carrier to Noise Schedule (NCS) introduces fairness into the scheduling by selecting the user with the highest normalised CNR at each time slot (Yang & Alouini, 2004). Multiuser diversity is inherent within NCS due to the fluctuations of the channel conditions. By scheduling the highest ratio CNR given by $\mathrm{X_i}\,(t) = \frac{\gamma_i(t)}{\bar{\gamma}_i}$ (Hassel V. , 2007), a multiuser diversity can be maximised. This is an instance of the CNR level of user i divided by the average CNR. The average is calculated over a weighted time window. This algorithm is not adaptable to fluctuations in the CNR. When always scheduling the users with the highest CNR means users will be close to their peak CNR value before they are selected. With NCS each user will have a multiuser diversity independently and identically distributed with the same average CNR (Hassel, Alouini, Gesbert, & Øien, 2005).

2.13.4 Opportunistic Round Robin (ORR)

Kulkami and Roseenberg (2003), developed Opportunistic Round Robin (ORR) technique to increase the short term fairness between users (Kulkarni & Rosenberg, 2003). The time slots are allocated as N time slots, where N represents the number of current users. The user with the highest CNR is allocated the first time slot. Once a user has been assigned a time slot it is then removed from the allocation process during that iteration. Implementing this achieves the highest short term fairness. The difference with ORR is that although users are assigned one time slot this is accomplished opportunistically which maximises the total throughput, within each iteration. Johanosson (2007) enhanced the algorithm to utilize the channel state information of the users in the next time slot. Here the channel conditions of the user are advantageously exploited while simultaneously ensuring that the allocated time slots are evenly distributed (Johansson, 2004). This algorithm depends on the differentiation of average CNR's to determine those with the highest CNR resulting When there is a wide spread of average CNR's, the opportunistic assignment of time slots are not evenly distributed.

2.13.4.1 Cross Layer Algorithms

Cross layer algorithms incorporate both QoS and opportunistic channel condition elements dynamically, with the aiming to provide the optimum performance. Hassel (2007) further breaks these down into non-queue aware and queue aware (Hassel V. , 2007). Non-queue aware does not consider the behaviour of the queue and its impact on delay, nor the actual channel conditions or the QoS prioritization, whereas queue aware incorporates queue behaviour when scheduling. Therefore non-queue aware algorithm can seriously impact on QoS and especially SF requirements within WiMAX. Therefore here the focus will be on queue aware scheduling.

2.13.4.2 Modified Largest Weighted Delay First (MLWDF)

The key features of the Modified Largest Weighted Delay First (MLWDF) algorithm is that the scheduling decisions depend on both current channel conditions and the state of the queue. The scheduler time stamps arriving data packets for all users, this enables tracking of the current queue length. For each user the latency distribution is controlled by the parameter γ_i of equation (2.11). Therefore if γ_i is increased for user i and the other users are unchanged the packet delay for user i is then reduced (Andrews, Kumaran, & Ramanan, 2001), enabling shaping of the delay distributions (Stolyar & Ramanan, 2001).

$$i * (t_k) = \arg\max_{1 \le i \le N} \gamma_i W_i(t_k) r_i(t_k) \qquad (2.11)$$

In equation (2.11) time slot (t) serves the flow i, where $W_i\,(t)$ is the head of queue packet delay for *I*, whereas $r_i(t_k)$ is the channel capacity for flow i. The queue status is monitored along with the channel condition which enables the algorithm to optimize the throughput of particular connections while ensuring the queues do not reach a fully congested state. It ensures the queue stability as long as the flow arrival rates are within the system stability region (Ohrtman, 2005).

It is assumed that this algorithm can be efficiently used in applications to provide a flexible control of QoS for multiple data flows especially when sharing a time varying wireless communication medium. Further extending the MLWDF algorithm if the OFDMA environment of the delay sensitive traffic is comfortably within its boundaries the priority constraints are relaxed. This implementation of the algorithm improves efficiency by providing transmission opportunities for less important priority queues which would initially be in a wait state until the delay sensitive traffic has completed its transmission. Max-Weighted scheduling algorithms such as MLWDF retain stability allowing more flexible control of queue length and delay distributions to satisfy a variety of QoS requirements. Those SF's within WiMAX that are not delay sensitive nor have good channel conditions could still suffer lack off transmission opportunity.

2.13.5 Hierarchical and Hybrid Algorithms

In the previous sections we have ascertained that scheduling algorithms achieve a variety of different entities of QoS, as scheduling services have different requirements. UGS prioritises the delay and bandwidth requirements, while rtPS and ertPS prioritise delay, especially eliminating jitter. But as nrtPS and BE is delay tolerant, it is susceptible to scheduling starvation, accommodating these SF is essential. Hierarchical algorithms use levels of decision making in the packet scheduling, whereas hybrid algorithms are made up of multiple scheduling algorithms to accommodate the QoS requirements of the traffic. For example EDF can cater for delay sensitive classes such as UGS, rtPS and ertPS, WRR can be called upon to maximise throughput as in nrtPS and BE. Hybrid algorithms can become hierarchical in nature.

Ramachandran, (2004), presented a link adaptive algorithm that utilizes the recommended standard measure of $C/(N+I)$, where C is the ratio of the received signal power, I is the power of the interference and N is the noise. The algorithm focussed on the downlink and assumes the uplink is transmitting satisfactory. Each SS continuously monitors its own measurement and changes modulations if conditions dictate. By defining the thresholds of each channel type, it ensures that the link adaption is fully optimized on the downlink (Ramachandran, 2004). Mähönen, Saarinen, & Shelby (2001) identified there is no automatic repeat request (AQR) scheme in lossy channels therefore sacrificing some bandwidth and increasing the physical layer robustness is actually more effective than using ARQ (Mähönen, Saarinen, & Shelby, 2001). Ramachandran, Bostian, & Midkiff incorporated TCP to manage delay, by encompassing the channel deterioration into the decision making. This then became the link quality indicator. By coupling TCP with a dynamic window scaling system to manage delay based on SNR which proved insensitive to the TCP improvements (Ramachandran, Bostian, & Midkiff, WCNC 2005, 2005).

Liu et al (2005) and Chen et al (2005) proposed an algorithm that was based on EDF and DRR. This multi-level scheduling algorithm is based on per connection to meet QoS requirements, which causes further delay in the form of overheads (Liu, Li, & Pei, 2005) (Chen, Jiao, & Wang, 2005). This is due to the translation of QoS requirements into each scheduler configuration at every level. The actual scheduling calculation is ongoing because the state of the system is in a continual variable flow, e.g the data rate changes constantly. To put this into perspective assuming a throughput of 400 frames per second (IEEE , 2004) of a multimedia application if the BS is then making all the scheduling decisions then there is 400 scheduling decisions to be made per second, therefore the overheads of such multilayer scheduling algorithms is great (Sayenko, Alanen, & Hämäläinena, 2008).

To maintain good QoS the system must be interoperable, able to accommodate of all types of categories of traffic, ensuring efficient bandwidth utilization, though it is recognized that this alone is a challenging undertaking. Chen et al, (2009), proposed the Variable Bit Rate Video Service

(VBRVS) method for scheduling video traffic (Chen, Deng, Hsu, & Wang, 2009). The VBRVS uses two mechanisms defined by two levels, One Level VBRVS algorithm and Two Level VBRVS algorithm. The One Level VBRVS algorithm involves the BS assigning uplink bandwidth to the users using video based on their traffic state transition; when the current state is changed, a bandwidth request is required. This algorithm accommodates variable bit rate traffic that may have peaks in the transmission stream. The constraint of this algorithm is the bandwidth request itself, which adds to the network traffic therefore it has the potential impact on delay. To achieve this, the traffic state needs to be monitored, when the resources are at capacity the BS needs to determine whether the request can be granted.

The second mechanism, the Two Level VBRVS algorithm allows the bandwidth request process to be reduced, thereby reducing the bandwidth usage. To achieve this, the algorithm uses two bits within the MAC header allowing the BS to know the demand of the SS without utilizing an additional request mechanism. The BS can then decide if a bandwidth request opportunity should be assigned, or a data transmission opportunity granted, providing a more deterministic bandwidth allocation mechanism. Chen, Deng, Hsu, & Wang (2009) results evidenced that both the VBRVS algorithms will have "*less bandwidth waste ratio than the conventional uplink scheduling algorithms*." (Chen, Deng, Hsu, & Wang, 2009).

Dai & Zhao (2007) research focused on mobile real-time traffic, particularly applications that supported realtime voice traffic . Their assumption was that the WiMAX IEEE 802.16 based backhaul network involve multiple BS's connected together using a microwave link (Dai, & Zhao, 2007). This enabled connections to be established whilst moving. Dai & Zhao (2007) suggests a "*simple enhancement to the bandwidth request mechanism in IEEE 802.16 for* *supporting packet voice traffic*" (Dai & Zhao, 2007). Each proposed scheduling scheme to-date will require a different amount of information to be included in the bandwidth request. In Dai & Zhao's (2007) scheme the bandwidth requests are aggregated. The aim is to make the bandwidth request process more efficient, by cutting the amount of time and bandwidth required by both the SS and BS for requests. These aggregate bandwidth requests will include information regarding the latency requirements of buffered real-time packets from the SS, this will help the BS to make informed resource allocation decisions. Through simulations it was evidenced that the system provided "*satisfactory real-time performance for voice traffic*" (Dai & Zhao, 2007). Results showed that "*there is an optimum amount of information to be transmitted in the bandwidth requests in order to achieve good voice packet transmission performance*" (Dai & Zhao, 2007).

Wongthavarawat and Ganz (2003) developed a hybrid algorithm with the aim to accommodate the QoS requirements for all service classes (Wongthavarawat & Ganz, 2003). Each SF has its own strict priorities for QoS, the highest being UGS followed by rtPS ertPS nrtPS and BE. Currently the issues with a single scheduler algorithm they do not accommodate all QoS requirements for all the SF's which can result in packets missing their deadline or the lower SF's priorities such as nrtPS and BE end up being starved of bandwidth. Wongthavarawat and Ganz (2003) have proposed a combination of scheduling algorithms to overcome this problem, by using traffic policing to ensure UGS has the fixed bandwidth as stated in the IEEE 802.16 standard while EDF can accommodate bandwidth allocation for rtPS. By using an information module determines the packet deadline. WFQ can schedule nrtPS based on the weight of the connection, while still allocating bandwidth as necessary to BE. Wongthavarawat and Ganz (2003) did not incorporate the whole algorithm in their simulated results and only focused on rtPS and BE, but this did evidence that these two service classes can be scheduled without any packets missing there delay deadline and providing adequate bandwidth to avoid starvation of the SFs (Wongthavarawat & Ganz, 2003).

Cao & Li, (2001), states that hybrid scheduling algorithms utilize different scheduling algorithms for different service classes but before selecting the appropriate scheduling algorithm the requirements of the queued traffic must be taken into consideration along with the complexity of the scheduling and to accommodate fair distribution of the resources among the SS's. Cao & Li (2001), proposed an algorithm that used WFQ using the connection weight to adjust to the priority needs, nrtPS applies Dynamic RR and the simple FIFO is deployed for BE (Cao & Li, 2001). Cao's algorithm did not include the details on how the mix of scheduling algorithms is controlled from the BS. Here additional system delay is inherent due to the overheads of this control, (Cao & Li, 2001).

Vinay et al (2006), states that using a hybrid-scheduling algorithms for real-time communications produce better end to end delay than using a single algorithm for all services (Vinay, Sreenivasulu, Jayaram, & Das, 2006). Vinay et al (2006), infer from their simulations that a dynamic approach to distribution of bandwidth improves the number of UGS connections in comparison to a single algorithm (Vinay, Sreenivasulu, Jayaram, & Das, 2006). Dhrona, et al (2007), scheduling algorithm encompassed EDF for ertPS and rtPS while employing WFQ for nrtPS and BE classes, concluding that scheduling needs to be selected based on the QoS requirements and traffic profiles (Dhrona, Abu Ali, & Hassanein, 2008). This is due to a mixture of traffic types with varying QoS priorities, Dhrona et al (2007), found that EDF algorithms are best as it caters for delay and packet loss while providing a high average throughput and fairness, but WRR did not cope with traffic with variable sized packets (Dhrona, Abu Ali, & Hassanein, 2008).

Dhrona's et al (2008) work encompasses CAC schemes coupled with scheduling algorithms with the aim to ensure the number of connections in the network can be served efficiently while providing satisfactory QoS. They state the choice of CAC scheme is critical for the performance of a scheduling algorithm, making it vital to evaluate the different scheduling schemes in relation to the CAC being used (Dhrona, Abu Ali, & Hassanein, 2008).

2.14 Summary

This chapter investigated two key aeras QoS and Scheduling. The IEEE 802.16 benefits from being able to deliver a connection oriented protocol which uses SF that represents the type of traffic that is being transmitted while still providing a scalable and reactive environment. The standard is compatible with the ATM QoS mechanisms. The grant-based MAC layer allows centralized control eliminates overheads and delay of acknowledgements. By aggregating the bandwidth requests the SS can react to the QoS needs in real-time reducing the computation that the BS needs to complete.

The IEEE 802.16 standard also encompasses OFDM and OFDMA to provide error correction and interleaving of the transmission to further improve QoS. Therefore the PDU's can be fragmented to enable larger SDU's to be sent across frame boundaries to guarantee QoS of competing services. Adaptive modulation corrects distance problems by enabling channel divisions based upon frequency division or time slots. The channel size also changes within the range 1.5 to 28 MHz spectrums. If the SS has no direct link to the BS, communication can still be maintained via other SS's using a mesh topology (Shetiya & Sharma, 2005).

The scheduling algorithm taxonomy categorized the algorithms into four sections: Balance fairness and throughput, weight/deficit calculations, opportunistic/crosslayer and hybrid/hierarchical. In the Balance fairness and throughput category, it was found that the RR algorithm allocated resources inefficiently resulting in reduced throughput, but it upheld the fairness criteria. Both Max SNR and Proportional fair algorithm had an imbalance of fairness but maintained good throughput. Where the Urgency variable incorporated into the proportional fair algorithm resolved the fairness issue, but was not adaptable to the channel conditions. Within the weight/deficit category a conflict between the algorithms weighting of the decision making variable and the actual transmission priority of the data was identied. In the opportunistic and crosslayer category scheduling decisions were based primarily on the channel conditions, therefore attempting to use the resources efficiently. But as the algorithms use valuable resources to monitor the condition of the channel, there is a trade-off between the scheduling gain and short term performance. From this category, only ORR manages to provide good short term fairness, but this deteriorated over time. With the cross layer approach MLWDF optimizes throughput while ensuring the queues do not get congested, it had good control of the queue length and delay distribution. To accommodate full QoS scheduling the Hybrid/Hierarchical categories use a combination of the algorithms to best suit the target environment as discussed in this section, but none of the algorithms address all the QoS parameters while still maintaining both priority and fairness. Dhrona et al (2007) infer scheduling and CAC are not mutually exclusive and both need to work together to ensure an efficient system.

Chapter 3 Call Admission Control (CAC) Schemes

The Call Admission Control (CAC), (see figure 4.1), role within the IEEE 802.16 standard is to regulate the traffic volume (IEEE Standard 802.16.2, 2004). The method to accomplish this is not currently defined in the standard therefore it makes sense to investigate this area especially as the CAC is an important part of QoS provisioning within the standard (IEEE Standard 802.16.2, 2004). As IEEE 802.16 standard operates over a connection oriented protocol, which means that a connection needs to be established end-to-end between the SS and the BS before the transmission of the data can commence. To initiate the connection, the SS sends a request to the CAC to establish the link. The request contains the required connections QoS parameters so that the BS can decide whether to 'admit' the connection or not. The BS needs to ensure it can provide the required QoS resources for that connection request, while continuing to maintain the connections previously admitted to the BS, before accepting the request. The main aim for CAC is to 'admit' connections while guaranteeing QoS for the connection, where the request can contain varying QoS parameter requests. While ensuring a decrease of the Connection Blocking Probability (CBP), the Connection Dropping Probability (CDP) and optimizing the Bandwidth Utilization (BU).

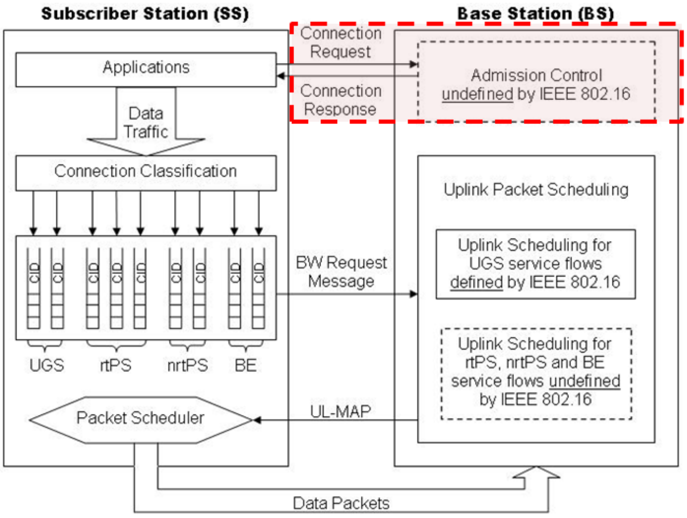

Figure 3.1 IEEE 802.16 QoS Architecture (IEEE Std 802.11e-2005, 2005)

3.1 Call Admission Control Provisioning

Many researchers have proposed enhancement to the CAC scheme, for example Wang and Li et al (2005) proposed a scheme that uses bandwidth reservation and degradation polices to deliver connections with QoS (Wang, Li, & Agrrawal, 2005). The bandwidth reservation prioritizes the requests for services flow such as UGS connections. It utilizes degradation as a method of decreasing the bandwidth allocated to the admitted connections in order to accommodate additional connections. MSTR and MRTR (see section 2.9.1, P71) provide the parameters to degrade the bandwidth. If there are only a few connections currently admitted then they are all allocated the full MSTR. As the number of connections increase the bandwidth allocated to the existing connections are decrease until it reaches the minimum level that the connection requires. But these parameters are only defined for rtPS and nrtPS so this algorithm can only be applied to these SF (Wang, Li, & Agrrawal, 2005). This degradation model is based on the IEEE 802.16d version of the standard, the aim of which is to improve the bandwidth utilization while decreasing the blocking probability of the services flows creating a constant rate CAC scheme. The constraint of the model does not provide any delay guarantees to the admitted connections. A comparable architecture was proposed by Chu, Wang & Mei (2002) work, though inclusive of a CAC, also encompassed some scheduling aspects within their model (Chu, Wang, & Mei, 2002). The results of the architecture were comparable with Wang's et al (2005), both with no guarantee of delay parameters.

Whereas Wang and Liu et el's (2007), work offered a prioritized handoff scheme based on the IEEE 802.16e version of the standard (IEEE 802.16e-2005, IEEE Std 802.16-2004/Cor1-2005, 2006) (Wang, Liu, ji, & Ruang-chaijatupon, 2007). The scheme prioritizes the handoff connections over new connection requests using the guard scheme which can result in an over allocation of bandwidth. The scheme encompassed a degradation mechanism for prioritizing different types of services which increases the performance of the system. This is achieved through the utilization of the guard channel and borrowing bandwidth from available resources to ensure that UGS, ertPS and rtPS SF are maintained. This scheme ensures that bandwidth utilization and priority of bandwidth allocation is maintained. The scheme reduces the CBP and CDP of each kind of connection. The disadvantage to the scheme is that it is unfavourable to nrtPS and BE SF as they can be starved of bandwidth with this scheme (Wang, Liu, ji, & Ruang-chaijatupon, 2007). Whereas the proposed resource allocation and redistribution algorithm (RHR-CAC) in this work considers the transmission opportunities of SF's while providing support for QoS (Chapter 6)

Ge and Kuo's (2006), work also utilize a method that prioritized handoff connections. This was achieved by allocating the surplus bandwidth of the admitted connections. New connections are only admitted if there is enough bandwidth to meet the QoS requirements without resorting to using the surplus bandwidth from other connections as a resource. The method focuses at the allocation of the bandwidth but does not guarantee the QoS delay parameters either (Ge & Kuo, 2006). The RHR-CAC algorithm will use surplus bandwidth of granted connections to improve connection rates while supporting SF priorities, this algoritjm is invoke only when the resource allocation of the BS reaches saturation point.

Research in QoS for the IEEE 802.16 standard is generally focused on either the scheduling or the CAC aspect of the QoS architecture, Wongthavarawat and Ganz (2003) were one of the first researchers to encompass both elements within their research. They used the CAC to provide the bandwidth guarantees for all the SF (except ertPS this was not included into the standard until 2005) while adhering to the delay requirements for only the rtPS SF. As UGS has automatic bandwidth guarantees using the request transmission policy and nrtPS and BE are not sensitive to delay each SF has a different scheduling algorithm applied to it. The order of priorities adopted is UGS > rtPS>nrtPS>BE (Wongthavarawat & Ganz, 2003). As can be seen from Figure 4.2 each SF has a discrete scheduling algorithm associated to it, these are invoked in order of the strict priority.

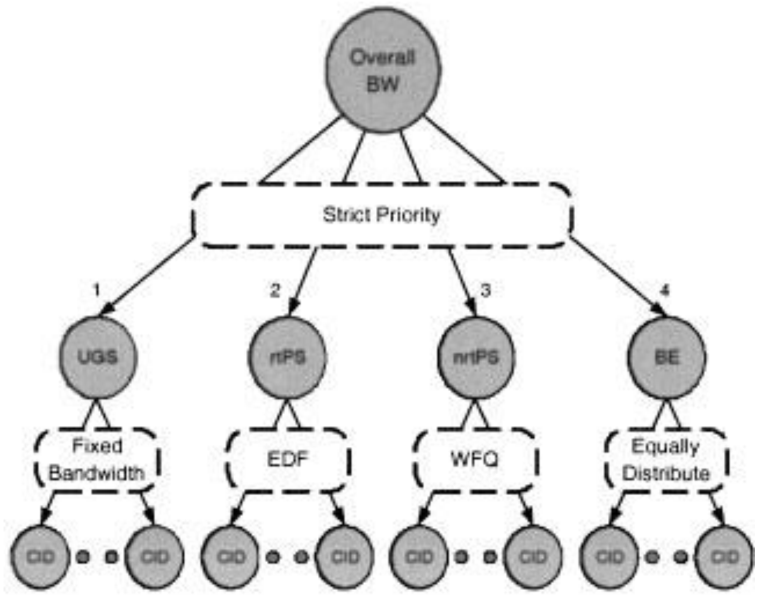

Figure 3.2 Hierarchical structure of bandwidth allocation (Wongthavarawat & Ganz, 2003)

The CAC ensures that each existing connection has the requested QoS resources (such as required latency tolerance) available to it ensuring there is enough bandwidth and the QoS levels are guaranteed and can be maintained for each connection. A traffic policing module ensures there are no QoS violations. If there are QoS violations detected then the resources are adjusted to attempt to rebalance the QoS requirements. At this point all new connection requests are rejected, until all violation has been rectified. Wongthavarawat and Ganz (2003) focused on the rtPS and BE only, their results do evidence that all rtPS packets met their deadline or BE had adequate bandwidth to maintain the required transmission of packets (Wongthavarawat & Ganz, 2003).

Jiang and Tsai (2006) also incorporated both CAC and scheduling in to their algorithm, that supports QoS via uplink packet scheduling and the CAC mechanisms, but without degradation. Again the bandwidth is reserved for rtPS SF while the scheduling algorithm aims to meet the delay requirements, the results for this work were generated using Poisson traffic generation and converting it in to a token bucket based connection. Both proposals improved BU, CBP and CDP (Jiang & Tsai, 2006), but not all SF were tested only rtPS and nrtPS.

Kalikivayi et al (2008), built upon the work of Wang and Liu (2007) and Wongthavarawat and Ganaz (2003), to improve the bandwidth and delay guarantees to the rtPS SF, by modifying Wongthavarawat and Ganz (2003) CAC algorithm to meet the bandwidth and delay requirements encompassing an analytical model based on the Continuous Time Markov Chain model as used by Wang and Liu (2007) (Kalikivaya, Misra, & Saha, 2008). Kalikivayi et al (2008) assumed the rtPS and ertPS are equivalent and the QoS parameters only differ in the request/transmission policy, also BE was not considered in the algorithm (Kalikivaya, Misra, & Saha, 2008). The availability of the QoS requirements are checked for their availability before the connection is admitted if the resources are not available then the connections are rejected. Kalikivayi et al (2008) method improves CAC in terms of CBP and the handoff CDP of different types of connections and BU of the system. The results also illustrated that the threshold limit for delay guarantee is obtained.

Tsang et al (2007), developed a method where the CAC blocks unwanted connections in order to maintain the QoS of existing connections and reduces the buffer needed for packet scheduling (Tsang, Lee, Tung, Lam, Sun, & Ko, 2007). The Quadra Threshold Bandwidth Reservation (QTBR) uses a first come first serve on the uplink packet scheduling system utilizing GPC mode. The admittance policy of the CAC is a trade-off between the CBP/CDP together with the end to end delay of each connection. When heavy traffic loads are present a control factor is used to determine the allocation of the bandwidth, if the traffic load is heavy the allocation of bandwidth to each connection is reduced to the minimum amount that would satisfy the QoS requirements to cater for additional connections.

CAC research is not independent but is closely coupled to scheduling efficiency to deliver satisfactory QoS to the connection. Wang et al (2005) and Chu et al (2002) who focused on CAC all found difficulty in meeting delay requirements, (Wang, Li, & Agrrawal, 2005), (Chu, Wang, & Mei, 2002). Degradation was utilized to maintain QoS for existing connections for both algorithms. Wongthavarawat and Ganz (2003) and Kalikivayi et al (2008) both used scheduling to assure delay requirements, though neither presented results including all SF. In all cases there is a balance between CBP and CDP, while maintaining efficient BU.

3.2 Call Admissions Control: The Constraints

CAC provides a vital role for QoS provisioning within the context of the IEEE 802.16 standard (section 2.4). Coupled with the MAC layer, the CAC facilitates a process that enables the end user's application to receive the required QoS to provide satisfactory connectivity. As the IEEE 802.16 standard is connection oriented, the whole CAC process needs to encompass QoS from the initial connection setup right through to the actual termination of the connection. Therefore the CAC process facilitates the establishment of the connection from the BS to the SS, and in doing so, the BS ascertains the availability of the required resources to fulfil reservation requests, before allocating the relevant SF. The SF provides a method to determine both downlink and uplink QoS management on an individual SS to BS connection basis. Fundamentally the BS needs to ascertain the availability of the requested resources before accepting the SS's request for a connection. The BS must manage its resources appropriately, ensuring that they are not over committed, as a connection is not accepted unless all the requested resources are available to be allocated to that particular connection request. Otherwise if the resources are not available the connection request is instantly rejected (figure 3.1 & 3.3).

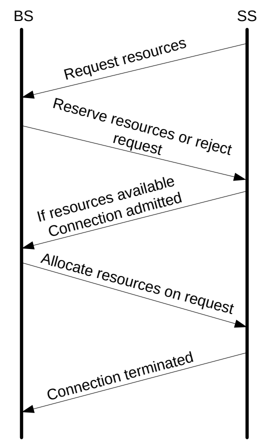

Figure 3.3 Overview of establishing resource reservation for a BS to SS connection

The CAC mechanism of the IEEE 802.16 standard is currently undefined (see figure 4.1), as explained in section 4.1. Though vendors such as WiBRO and Sprint have deployed WiMAX based infrastructures, the CAC element of the implementation is vendor specific. Therefore neither is interoperable, nor standardised, resulting in cross platform communication constraints (Farrell & Klemperer, 2007), which is a contradiction of the open standards and interoperability of systems that Sir Tim Berner-Lee advocates (section 2.2, p53).

The standard does include a QoS parameter set, which is available to the BS, these can be utilized to request resources, determine the current availability of the requested resources, and determine if they are available before reserving them for the SS connection. The parameter set includes ProvisionalQoSParamSet; AdmitedQoSParamSet; ActiveQoSParamSet, (see figure 2.13). These CAC QoS parameters are managed through the MAC Layer messaging system section 2.8.2. Initially designed for a static configuration between the SS and the BS, the current evolutions of the IEEE 802.16 standard incorporates a more dynamic approach to update existing connections (Meucci, Pierucci & Cerutti, 2010).

Coupled with the QoS parameter sets, the standard also defines the SF (UGS, ertPS, rtPS, nrtPS, and BE) mechanisms within the IEEE 802.16 standard (see section 2.7.1) which are fundamental to the MAC protocol. SF's are a defined QoS management mechanism provisioned on an individual BS to SS connection basis in turn enabling the BS to manage all current connections. Once the resources have been reserved via the QoS parameter set, the SF defines the service type priority that the connection has been allocated, in turn determining the level of QoS required.

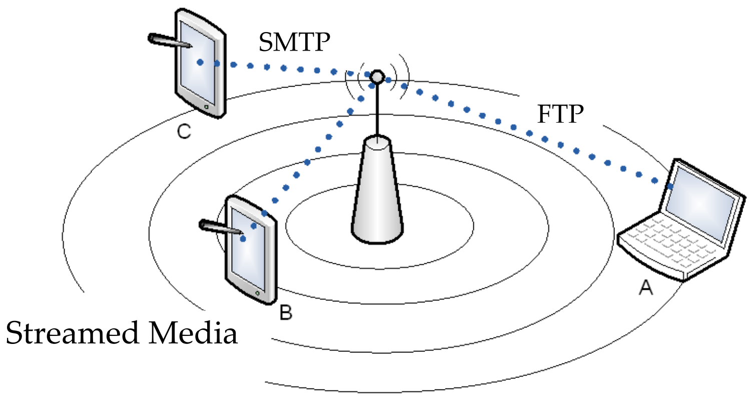

Figure 3.4 Example of BS to SS Communication Based on Traffic Types

Figure 4.4 illustrates a scenario where three devices request a connection from a BS. If the connection requested from all three devices happen simultaneously how does the BS manage the requests? How will the BS prioritise the requests that they receive? If the BS schedules the requests via a

FIFO method (section 2.10.1) the resource requirements are not being considered, nor are the QoS priorities of the connection. For example, if device A's request is serviced first (being first in the queue) but the resource requirements are for a file transfer protocol (FTP) connection, which has a low QoS priority. But, by servicing this request first and allocating the bandwidth required to device A, the BS cannot now supply the requested resources for device B that requires enough bandwidth to support a streamed media application such as youtube©. In this scenario the connection request would then be refused. Then if device C's request is serviced next and requires resources for a Simple Mail Transfer Protocol (SMTP) connection, to check email, the BS may still have enough resources available to grant this connection. By utilizing the FIFO scheduling algorithm the higher priority request is refused and the lower priority requests are serviced first. The assumption is that the FTP and SMTP requests could wait longer for their resources to be allocated, as they come within the BE SF category (section 2.9, p71), which has the lowest priority. Whereas device B's streaming media resources should be allocated the available resources first as it has a higher priority e.g. ertPS or rtPS, rather than just for those allocated a FIFO priority. The problem is that the BS needs to determine these priorities within its admissions policy to ensure QoS for all end users and reduce the quantity of rejected connections, while also ensuring that the lower priority SF's are not starved of connection opportunities. To facilitate this, the BS will need to determine priorities before the actual resources are reserved ensuring that the higher priority connections will be serviced first. But this could still potentially starve the lower priority connections therefore it is vital the BS needs to be aware of all the priorities of each SF's connection requests e.g. latency requirements. Therefore by utilizing a weighted scheduling method (section 2.13.2, p90) to determine which connection should be allocated resources first based on QoS priorities would deploy resources more efficiently (Parekh & Gallager, 1993).

The basic fair queuing scheduling algorithm (section 2.13.1) reverts to FIFO along with the shortcomings as discussed previously. Whereas MAX CNR achieves maximum data throughput by utilizing the best channel rate subsequently starving connection opportunities to those with weaker channel conditions. To achieve a fairer throughput the CNR is averaged for each channel to negate connection starvations by exploiting the PFQ algorithm. But neither the PFQ (section 2.13.1.3, p85) nor the Max CNR algorithm (section 2.13.3.2, p101) supports QoS priorities. Therefore it is essential to differentiate each connection's request by providing a weighting as part of the decision making to support QoS. WFQ (section 2.13.2.4, p94) facilitates SFs priorities as part of the resource allocation, as well as the capabilities of providing bandwidth allocation to individual sessions. Stiliadis and Varma (1998) identified that WFQ can serve connections proportional to reservations while being able to distribute unused bandwidth among the active requests (Stiladis & Varma, 1998). When selecting a scheduling algorithm it is crucial to ascertain its implementation complexity, this should be independent of the quantity of admitted connections and balanced with the fairness it delivers. For example WRR has a complexity of 0(1) but as it is a linear algorithm consequently, the delay bounds grow accordingly. Whereas the complexity of the insertion/selection process in WFQ is 0(log N) which balances the latency with the connections in turn supporting QoS.

Figure 4.6 illustrates the CAC framework within the IEEE 802.16 standard, via a flow chart, for setting up a connection between the SS and the BS inclusive of the required resources necessary to achieve the appropriate level of QoS. The process commences with the SS scanning (1) for a BS that is in range to enable a connection (2). Once communication between the SS and the BS has been established, the SS will send the BS its QoS resource requirements for that connection this would include parameters such as bandwidth and latency requirements (3), upon receiving the QoS request the BS will check for the availability of these resources from those it currently has provisioned (4). If the resources are available (5) then they are admitted or reserved for the SS to utilize on that connection.

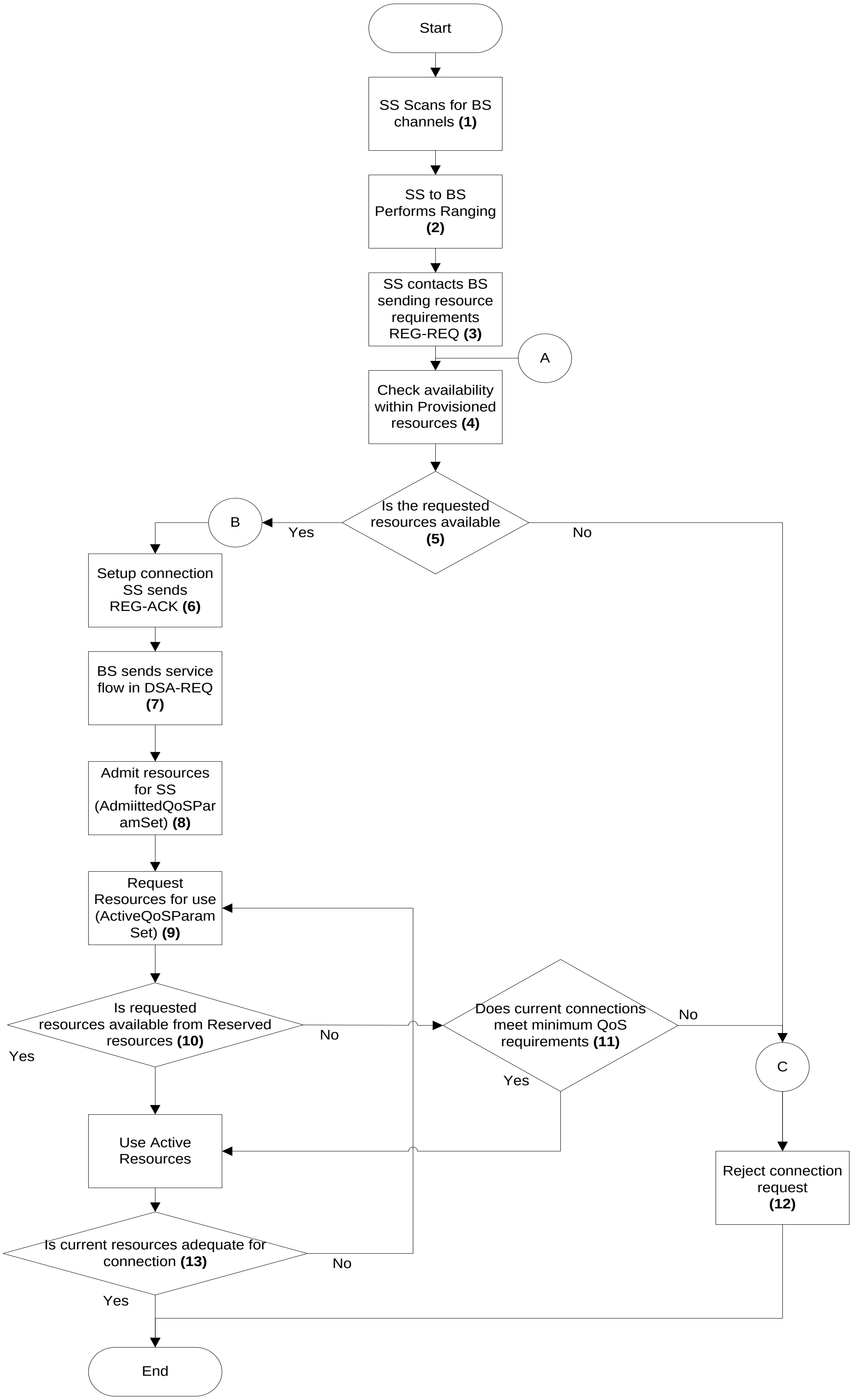

Figure 3.5 Call Admissions Control Algorithm

The connection is then setup and the SS acknowledges the acceptance of the connection via a REG-ACK message (6). Once the availability of the resource has been established the BS assigns the SF via a DSA-REQ (7), even when the communication and resources have been allocated but there are no guarantees that the SS will actually use the connection (8). When the SS decides to utilize the connection it sends a request to activate the required resources that has been reserved for it from the admitted resource allocation (9). The SS may not request use of all the resources initially. To terminate a connection a specific request needs to be made. Therefore it is possible for the connection to be granted but remain dormant for the duration that the SS remains in range of the BS.

A key feature of this mechanism is that the maximum available resources are only those that have been previously reserved in the AdmittedParamSet and no additional resources are made available beyond this (10). Therefore the SS can only request up to the limit originally stored in the AdmittedQoSParamSet (8), at which point if the QoS deteriorates to the point that it inhibits the current use of the connection (11) then the connection may be dropped (12). If the currently activated resources are below that which has been reserved (13) in the AdmittedParamSet (8) then additional resources can continue to be requested until all reserved resources are allocated to the connection are activated. If at the beginning of the algorithm the resources are not available the connection is automatically rejected without any compromise (12). The management of the downlink element of the connection is maintained through the BS via the QoSParamSet. It is the SS that will manage the uplink part of the connection.

Once the connection is granted it needs to be maintained, this is where the QoS messaging system can be utilized to provide dynamic configuration. This is based on the type of SF allocated coupled together with the applications being used. A change to the originally granted resources could be due to the application requiring more bandwidth to maintain its QoS to the end user. This could be peaks in transmission caused by streamed media e.g. fast action frames in films, or a VOIP call has become a group conference call, etc.

The disadvantage of this algorithm is threefold; firstly, once the resources for the connection have been admitted to the connection it maintains this reserved status even if the SS does not activate them while the SS is within range of the BS. Secondly, if the connection requires further provisioned resources over what has been currently reserved for that connection it cannot request supplementary resources above those that have been reserved in the AdmitQoSParamSet. Thirdly, this leads to an inefficient use of available resources, where connections could be rejected because there are no available provisioned resources even if there are still resources available that are currently not being utilized within the AdmittedParamSet as identified in Figure 4.4. Potentially within this algorithm there will always be resources allocated but not being utilized. Therefore, to improve the efficiency, the system needs to consider connections that either requires more resources or admitting new connections that will potentially be rejected due to lack of available resources.

Consequently this work investigates how to exploit those resources that are not currently being utilized. Though, in doing so, it is imperative that adequate QoS is maintained with the current active connections. Also the lower priorities of SF must still be considered for resource allocation, as prioritising connections only by the required SF will result in less resource hungry connection requests being rejected. Resulting in SF's such as nrtps and BE being starved of resources as discussed in the previous example figure 4.4.

3.3 Summary

The main aim of the CAC is to provision resources for the connection to ensure the required QoS levels per connection. Most research has been focused at either the scheduling or the CAC but Wongthavarawat and Ganz (2003) the first to encompass both elements, though they did not test their algorithms with all SF's. CAC only methods have covered algorithms such as Wang et al's (2005) reservation and degradation algorithm which aims to ensure the connections priority by allocation of bandwidth to accommodate the new connection. Chu et al (2005) proposed a similar algorithm but neither guaranteed the delay parameter. Wang et el (2007) utilizes a similar algorithm where the priority is the hand off connections over new connection requests, which improved performance. Ge and Kuo (2006) utilized the surplus bandwidth from the admitted parameter of the connections, rather than reducing the Active bandwidth of connection. But the commonality in many cases is the nrtPS and BE SF connection requests can be starved of bandwidth. The resources harvest and redistribution CAC algorithm (see Figure 4.5), incorporated features to serve all of the connection priority requests so that BE SF's are not starved of a connection opportunity, via harvesting redundant resources and distributing them to new connection requests.

The following chapter will outline details of two pilot studies. These will outline the viability of harvesting bandwidth to deploy to new connection to improve connectivity, and secondly smoothing the transition of moving between different strength signals.

Chapter 4 Pilot Scenarios: Resources Redistribution

This chapter serves to outline the viability of harvesting bandwidth to deploy to new connection to improve connectivity. Two pilot studies were carried out to firstly test the viability of redistributing bandwidth access to improve connection rates and secondly to predict network bandwidth fluctuation to enhance video stream service quality in the mobile domain.

4.1 Pilot study 1: The viability of redistributing bandwidth access to improve connection rates in WATM

Initially a pilot scenario was developed to test the viability of redistributing bandwidth access to improve connection rates. This was to test whether restricting bandwidth from current connections to enable new connections that would normally be denied due to lack of bandwidth availability will impact on overall QoS provision. As QoS is simply a mechanism for provisioning key resources, such as bandwidth, over a computer network for high demand content. The assumption is that the network can tolerate the harvesting of bandwidth to increase connectivity, within the 10% tolerance, (Kappler, Fu, & Schloer, 2011). For example, a simple home network may consist of an upstairs PC, and a downstairs living room TV, wirelessly connected to each other. The user may want to watch a digitally stored movie, located on the PC, on the downstairs TV without copying and storing the movie locally. First, the data must be streamed from the PC to the TV. Current streaming technologies (e.g. Windows Media 12 and RealVideo 10, introduced in 2011) permit near-DVD quality streaming at a rate of 500Kbps. This chapter focuses on a lower resolution form of video, approx. 320x240 with a frame rate of 20fps (frames per second) (A Review of Video Streaming over the Internet n.d.). This resolution is the minium standard resolution that popular video-streaming websites such as YouTube utilise (YouTube, 2015).

At this resolution and frame rate, video requires around 350Kbps to stream (Windows Media Encoded).

At these speeds, an IEEE 802.11b network (11Mbps) could sustain 31 simulations streams in optimal conditions, providing all streams were running a constant rate of 350Kbps. This does not include any network overhead, or any extrinsic factor affecting the quality of the network connection. The aim of QoS in this situation is to ensure each stream receives as close to 350Kbps as possible, and to insure a seamless change of bandwidth allocation to the user when necessary. The goal is to provide a dynamically changing wireless network (that is, a wireless network with devices leaving and entering the network on a ad hoc basis), with the ability to sustain every connected device running streamed multimedia applications with the required bandwidth and network conditions for, satisfactory use. The principle scenario would permit the complexity of multiple devices to simultaneously steam media.

4.1.1 ATM Networks

ATM (Asynchronous Transfer Mode) networks are combination of two commonly used network architectures, combining a packet switching network with a circuit-switching network employed for telecommunications (see section 2.2).

Although ATM networks have clear advantages, they have not been widely deployed in enterprise networks, aside for use in backbone technology. ATM is used for large-scale carrier networks as it provides good QoS. Classes of QoS support provided by ATM are very similar to those that are found in the WiMAX standard, it also utilizes a connection oriented protocol

ATM's cell structure is comprised of 53 bytes, of which 48 bytes are reserved for its payload. The cell incorporates a field for denoting the class of support that is currently in use (listed above). It is interesting to note (while discussing multimedia applications) that the development of the MPEG-2 standard was guided by the ATM architecture. MPEG-2 streaming is delivered in 188 byte sections, which precisely fit within the AAL5 layer of ATM. This development allows MPEG-2 streaming (up to DVD quality video) to take full advantage of the QoS provisions of ATM.

4.1.2 Wireless ATM (WATM)

Fundamentally the integration of ATM wired networks, into wireless platforms to encompass the end-to-end wired ATM advantages (namely QoS) in wireless technology. Wireless ATM architecture comprises of a collection of base stations interconnected using wired ATM technology to provide maintainable end-to-end communications, utilizing wireless technology.

4.1.3 Enhancing QoS for Wireless

One possible solution is to combine ATM QoS advantages with wireless adaptability to address the complexity of the problem, *Mobiware,* currently being researched at the Centre of Telecommunications Research, Columbia University, New York, attempts to do this.

4.1.3.1 Mobiware

Mobiware, is based on the latest distributed system technology and is a highly programmable middleware platform designed to run between the radio link layer and application layer of future next-generation wireless systems, such as base stations and WATM switches. Built on distributed systems and Java technology, it uses adaptive algorithms to transport scaleable transmission flows.

A very interesting aspect of *Mobiware* is its application specific 'flow adaptation policy'. This policy in its basic form characterises each transmission stream (flow) of data and recognises its acceptable minimal level of QoS. Using this information, *Mobiware* is capable of scaling each stream to match the available bandwidth while attempting to ensure that each transmission stream at least maintains acceptable QoS levels. A further claim of *Mobiware* is the provision of QoS support that allows multimedia applications to operate transparently during handoff and through heavy QoS requirement fluctuations. Figure 5.1 shows the architecture makeup of *Mobiware* (Campell, A. T., (n.d).

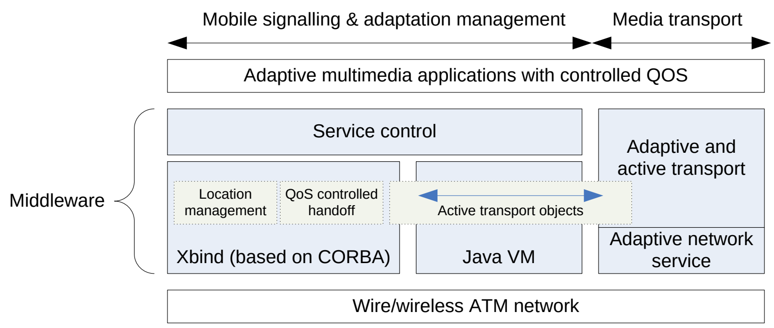

Figure 4.1. Mobiware Architecture, (Campell, A. T., (n.d))

Mobiware has set out to improve on multi-rate multimedia connections. These are difficult to achieve with current widely deployed technology, namely the Internet. With the rapid continual expansion of the Internet and the increasing demand for this type of usage, solutions must be found. Multi-rate connections also require management to ensure a seamless change in transmission quality to the end user. *Mobiware* achieved this through end-to-end QoS control using two methods: resource binding between devices and *Mobiware's* adaptive algorithms as illustrated in figure 4.1, (Campell, A. T., (n.d):

QoS controlled handoff: provides signalling of handoff events, used to represent flows, aggregation of these flows to/from devices and re-routing negotiation.

Adaptive network service: providing QoS guarantees based on available resources.

Adaptive and active transport: supports multilayer transmission flows via *Mobiware's* API.

4. 1.4 Intelligent Adaptive Buffer Control an Extension to Mobiware

Mobiware has begun to address QoS within wireless networks. QoS has limits and can only be provided when the available bandwidth exists, or, can be manipulated by QoS methods. However, can QoS be maintained in highly saturated networks with additional requests? A proposed solution for maintaining QoS in highly saturated networks, whilst still maintaining the levels of QoS for current users, but also providing the service for additional requests at the cost of initial delay is investigated below.

4.1.4.1 Intelligent Adaptive Buffer Control (iABC)

Intelligent Adaptive Buffer Control (iABC) is proposed to advance and build on *Mobiware*. The advantage of Mobiware's is its ability to scale transmissions for current available bandwidth while maintaining QoS. If *Mobiware* can indeed achieve this successfully, iABC would theoretically extend this concept. All technologies looked at so far are for network awareness, and low-level device awareness. Therfore these concepts will be transparent to the end user. iABC incorporates a high level of integration between the low-level network protocols (such as TCP and IP) and the end, high level user applications. The aim is to maintain QoS within a saturated network, while providing ad hoc bandwidth for additional applications. Application intelligence is required for this, and *Mobiware's* “flow adaptation policy”, if extended, could achieve this.

Technologies such as *Mobiware* try to guarantee QoS through secured bandwidth. However this is, a finite resource. Current video and audio streaming applications already use buffering to smooth any transitions in QoS that cannot be handled within the network. iABC would dynamically alter the size of the receiving applications buffer once additional demand occurs.

4.1.5 Bandwidth Distribution Scenario

In a home wireless network scenario example, that is capable of providing 800Kbps suppose that two nodes can stream video at 350Kbps. Alice (U_1), in the living room is watching internet TV while Bob (U_2), in the study, is watching a live news broadcast. Discounting any network overheads and assuming optimal conditions, there is currently 100Kbps unused ‘waste’ (it cannot be used for a third stream). Charlie (U_i) now requests a short 1 minutes “football update” to a mobile PDA. She requires the same bandwidth as Alice and Bob (350Kbps). The aim is to provide service for each device, despite the lack of available bandwidth whilst ensuring the QoS is not affected for any device.

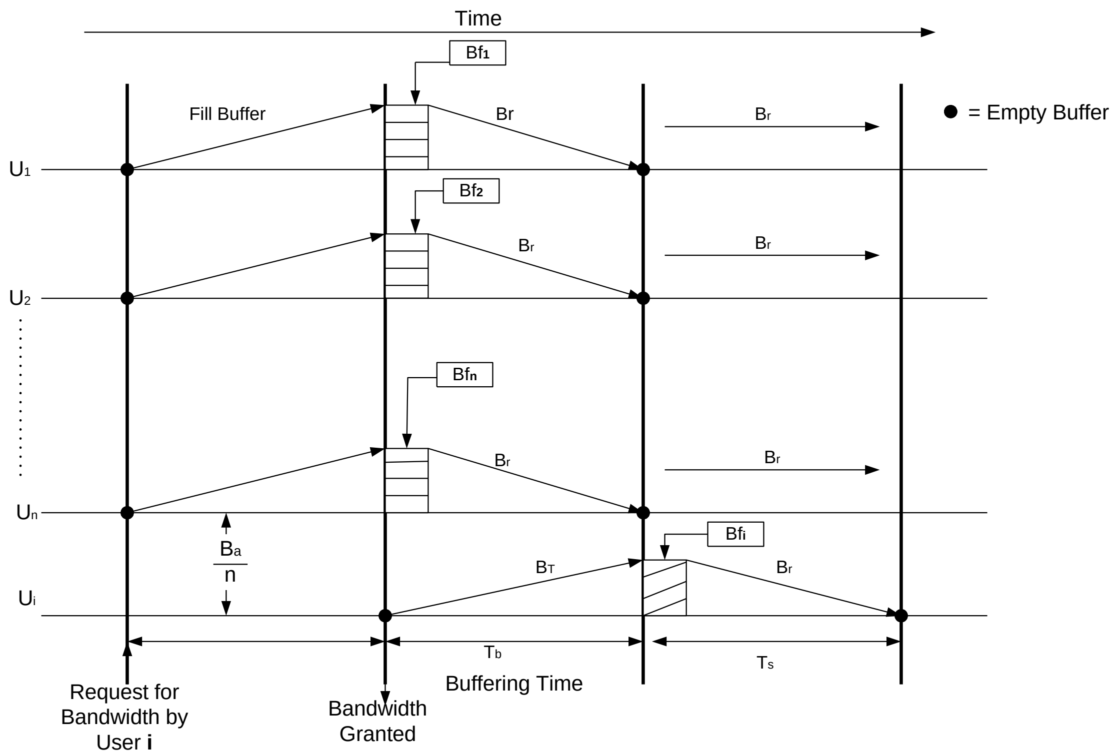

Figure 4. 2. An Example of iABC (Adda, Peart, & Watkins, 2006)

Figure 4.2 illustrates the above scenario in action, demonstrating the activities of iABC. Here, two users U_1 and U_2 are maintaining a connection for Mpeg2 video, streaming at 350kbps, without using any active buffering. A total bandwidth of 800kps is available, leaving 100kbps of bandwidth currently unused (B_a). This is determined (see equation 4.1) by subtracting the total bandwidth (B_T) available from the number of current users (n) multiplied by their bandwidth requirement (B_r)

$$B_a = B_T - nB_r \qquad (4.1)$$

A third user (U_i) requests to join the network, it is predicted that the request is 1 minute of streamed video requiring a bandwidth of 350kbps. It is also predicted that users U_1 and U_2 will demand their current level of bandwidth for a prolonged period of time, resulting in the user (U_i) request normally being rejected, which is unacceptable.

iABC proposes to dynamically buffer the requirements of the current users while maintaining the QoS, enabling the request of user (U_i) to be fulfilled. This is accomplished by utilizing the predicted requirements of user (U_i). Therefore determining users U_1 and U_2 buffer (B_f) requirements, B_f is calculated by multiplying the predicted transmission time (T_p) requested by the available bandwidth shared between the number of existing users (n) as shown in equation (4.2).

$$B_f = T_p * \frac{B_a}{n} \qquad (4.2)$$

This is also equivalent to the delay incurred before any bandwidth is available to fulfil U_i request. In this scenario it is known that U_i requires 1 minute at 350Kbps. Therefore if U_i could utilize the total bandwidth 26.25 seconds will be needed to completely buffer the requested video download bandwidth (Adda, Peart, & Watkins, 2006).

$$B_T = \frac{Bf_i}{B_t} = \frac{Bf}{B_r} \qquad (4.3)$$

The iABC instructs both users U_1 and U_2 to fill their respective buffers to hold the required data freeing up bandwidth for user U_i to initiate its request. Both users U_1 and U_2 share the remaining bandwidth to populate their buffers, the amount of data required in these buffer's is determined by the time taken to populate user U_i buffer as illustrated in equation (4.3). In this case given the available bandwidth per user is 50kbps, therefore in this scenario it would take 183.75 seconds to fill users U_1 and U_2 buffers. On completion the total bandwidth is release to user U_i to fulfil its request, therefore 21000kbps of data is buffered to provide a minute of video. Simultaneously the buffers of users U_1 and U_2 are being emptied, at the point when user U_i request is completely downloaded, users U_1 and U_2 buffers will be completely empty, at which point they will reclaim their original bandwidth requirement form user U_i, who now can access its data from the buffer.

iABC provides an interesting proposition to provide all parties with their respective requests, while maintaining a high and maintainable QoS for all. It did however, come at the expense of delay to the required service request. It is a high cost, but opposed to outright refusal of service could possibly be viable (Adda, Peart, & Watkins, 2006).

$$T_p = \frac{nB_a B_r^{\ 2}}{B_T(B_T - nB_r)} \qquad (4.4)$$

Combining all the variables into a single equation (see equation 4.4) further analysis can be carried out into the impact of varying demands on the bandwidth. As illustrated in figure 4.3 the longer the transmission period (Tp) the greater the delay is before the request is fulfilled (Adda, Peart, & Watkins, 2006).

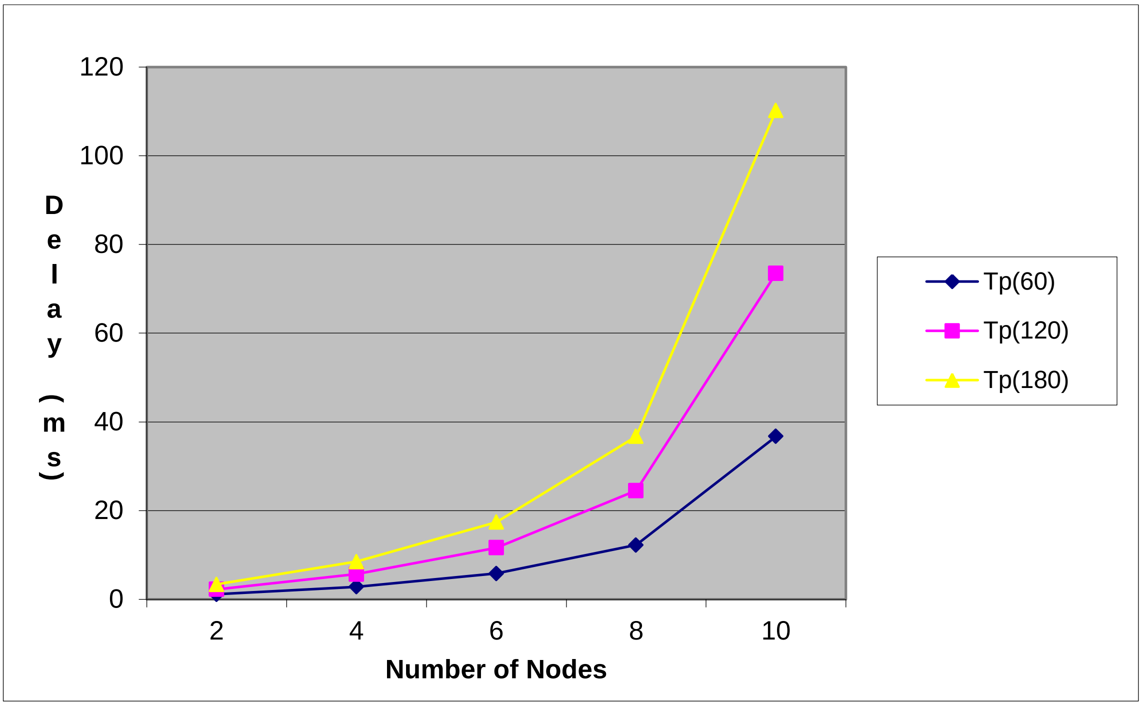

Figure 4.3. Effects of Transmission Periods on Delay (Adda, Peart, & Watkins, 2006)

There are two main drawbacks that are apparent. (1) Large lead-time for user U_i, before requested resource becomes available and, (2) iABC requires high-level integration with the end application to request and instruct buffer changes.

4.1.6 Summary: Pilot study

Today's common, wired network has achieved QoS, although limited by available bandwidth. Development is ongoing, and ATM networks are still immature other than backbone technologies. With the introduction of wireless networks, a new domain has opened with new challenges for achieving true QoS.

Mobiware is a software solution to QoS in wireless networks, attempting to resolve the problems and complexities, from current research; it appears to be making steady inroads. The proposed solution iABC demonstrates a viable solution, as illustrated in the example, where, two data streams tied the wireless network for a long period of time. In this circumstance, a 3 minute wait provided the requested resource as fast as possible in the given conditions. However, in different scenarios, if iABC incorporated further intelligent time slicing of the bandwidth between the devices the current delay would be improved. With the introduction of IEEE 802.11n wireless standard, and now at 1300Mbps, many more applications can be supported. Despite all solutions, QoS methods, additional bandwidth etc, the challenge is to try to make a resource that is finite appear infinite. Ultimately, bandwidth can only support a certain number of devices at a certain level of quality. When that limit is reached, additional requests will simply have to wait for freed resources.

4.2 Pilot Study 2: Predicting mobile network bandwidth fluctuation to enhance video stream service quality

Internet video is hugely popular across all devices – increasingly so on mobile devices. YouTube reported more than 1 billion users, with 300 hours of video uploaded daily in 2015, over 60% of which were on mobile devices outside the home (YouTube, 2015). In addition to Smartphone's, tablets have seen a large uptake by consumers in recent years with the success of Apple's iPad, and various manufacturers' equivalents. Even today's vehicular technology incorporates video streaming enabled devices as standard on most models of cars (NewComb, 2012)

This means physical movement from a city to a remote area often causes a sudden and significant decrease in available network bandwidth after network tower handover. This type of movement will be defined as as the movement from a Strong Signal Area (SSA), to a Weak Signal Area (WSA).

Streaming video over the Internet often relies on (pre-) buffering to allow for simultaneous download and playback. In order for a stream to be successful, the download rate must be consistently equal to, or mostly greater than, the video bit-rate. Whilst streaming video over a mobile data connection, moving from a SSA to a WSA causes severe bandwidth fluctuation – directly affecting the download rate. If a video data buffer cannot sustain an equal or greater download rate to the bit-rate of the content being watched, eventually the stream will be starved of data – causing a pause or pixilation of the video stream (Figure 5.4) Moving from a SSA implies that a high bit-rate video will be streaming to a user's device – so an unforeseen move to a WSA could cause video playback to pause abruptly.

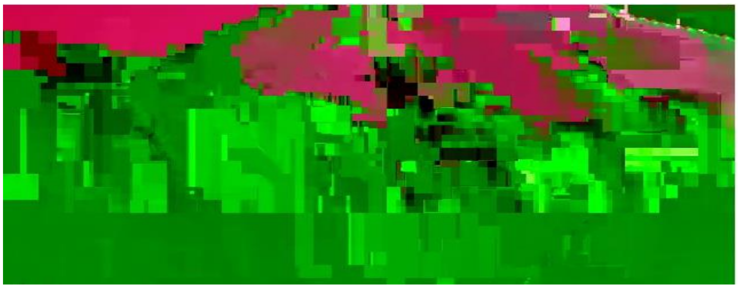

Figure 4.4 Mobile video stream pixilation due to being starved of data

Adaptive bit-rates have become the norm for many streaming services, with the standardization of the Scalable Video Content (SVC) MPEG-4 (H.264) extension. The bit-rate adaption allows for an appropriate quality of video to be chosen relative to the available bandwidth. SVC is exceptionally useful and effective for showing the same video in different qualities to viewers with high or low bandwidth availability, without needing numerous separately encoded video files (Ozer, 200). Tappayuthpijarn, Stockhammer and Steinbach, defines a HTTP-based video-streaming service that uses client-side algorithms to determine and request the "best", or most appropriate, quality for the currently available network bandwidth throughput (Tappayuthpijarn et al, 2011). Simulations of the proposed algorithm showed success with varying mobile network bandwidths. When mobile network bandwidth becomes restricted, the request for the SVC begins to omit layers to lower the bit-rate in an attempt to ensure the bandwidth is equal to, or greater than, the received bit-rate. However, regardless of how effective the algorithm is for handling near-real-time bandwidth fluctuations, the solution will still suffer from a poor QoS when a *sudden* bandwidth drop occurs. Moving to a WSA would initiate a request for lower bit-rate content by dropping content layers as described in the algorithm – but if the mobile signal is rapidly lost in its entirety, the request will be delayed or could already be too late. Both cases would rapidly lead to starvation of the existing stream buffer, and thus a pause in playback without any real attempt to prevent it ever materializing. This problem is addressed by the proposed solution detailed in this section.

Singh, Ott and Curcio, defines a new video stream service that this section shares some similarities with (Sing et al, 2012). Singh, Ott and Curcio, proposed service queries a “new” MNCM (this work does not include MNCM within its scope) to predict upcoming congestion within mobile networks to request bandwidth boosts from the stream provider. Singh, Ott and Curcio solution assumes the service reserves bandwidth for transmission rate increases to requesting devices (Sing et al, 2012). Reserved bandwidth is wasted when it is not being utilised. The location-based bandwidth prediction techniques are extended and adapted in this work to allow for dynamic data transmission rate-limitations within a video streaming service (Sing et al, 2012). The proposed techniques allow the service to impose temporary rate-limits on nearby “best-case” active users, removing the requirement for reserved bandwidth, or strict global rate-limits. Implementing this location-based dynamic rate-limit control to a video stream service could improve the QoS for users entering WSA’s, whilst also improving QoS globally by eliminating strict, consistent rate-limits. This section details the algorithms and services required to implement a dynamic bandwidth rate-limit control to a video stream service.

4.2.1 Maintain QoS when moving from SSA to a WSA region.

Predicting when a travelling user T_1 is about to leave a SSA and enter a WSA is fundamental to the solution proposed in this chapter. This solution assumes GPS information is available on the mobile device, and uses MNCM to predict upcoming constraints on mobile network bandwidth. The prediction algorithm locates best-case users B_k by analyzing other active user locations temporarily stored on the server. The service then applies a dynamic rate-limit to these users to allow for a bandwidth boost to T_1. Algorithms 1 (see Figures 5.5 and 5.6) and 2 (see Figures 5.7 and 5.8) defines how this can be achieved.

```
void GPS_OnDirectionChange(dir){
 C = GPS_getCoords();
 S = GPS_getSpeed();
 nWSA = getNearestWSA();
 if (directionLeadsToWSA(dir, nWSA)){
  reqBW = VideoSize−BufferedSize/(distanceToNearestWSA(C)/S)
  if (reqBW > currentRateLimit()){
   requestBoost(C, reqBW);
  }
 } else {
  stopBoostRequest();
 }
 sendCoordsToServer(C);
}
```

Figure 4.5 Client-side algorithm (pseudocode) to predict user entering a WSA

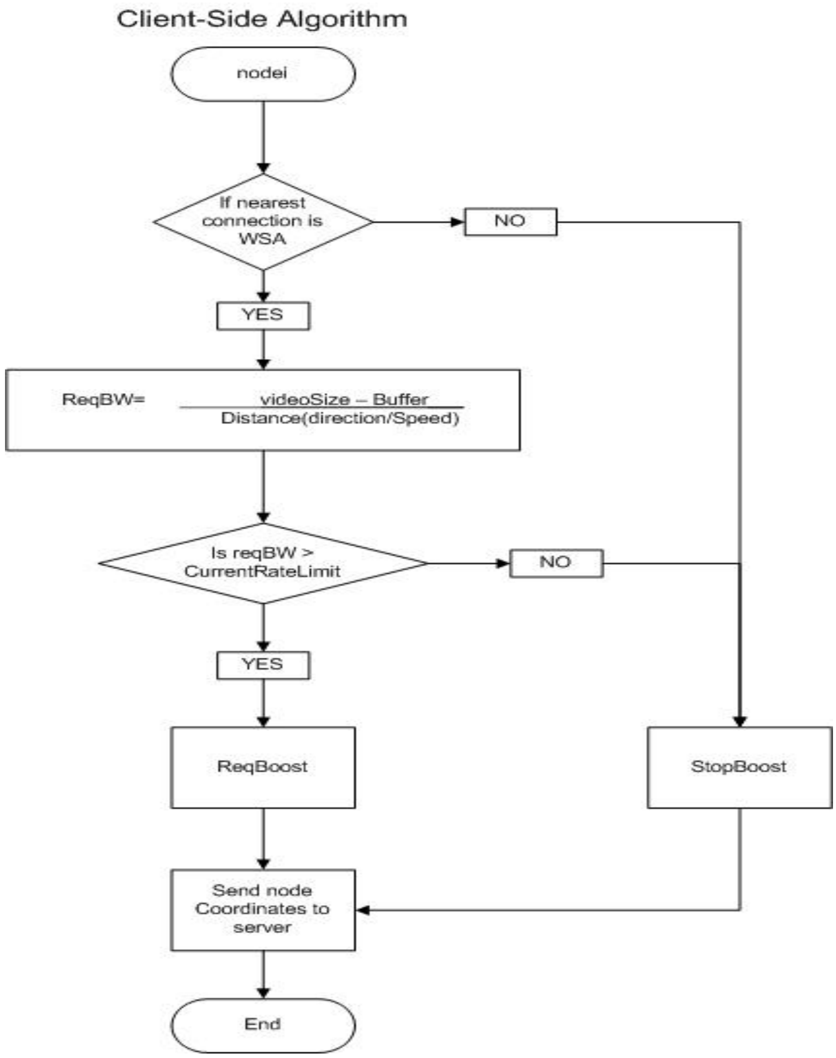

Figure 4.6 Client-side algorithm (Flow-Chart) to predict user entering a WSA

Figures 4.5 and 4.6 define an algorithm used to predict when T_1 is about to enter a WSA – taking speed and direction of travel into consideration. Algorithm 1 assumes a HTTP MNCM request and response as defined by Singh, Ott and Curcio is possible. In this solution, only network quality information for a given coordinate will be required from the MNCM – simplifying Singh, Ott and Curcio approach (Sing et al, 2011) The client-side algorithm (Figures. 5.5 and 5.6 assumes GPS functionality on a client's device).

A WSA should be flagged at a point of "low" mobile network bandwidth availability, rather than an area with extremely low e.g. GPRS, or no signal. Low signal areas, are classed as WSAs, used within *directionLeadsToWSA()* in Figures 4.5 and 4.6.

Figures 4.7 and 4.8 define an accompanying server-side algorithm used to determine which service users should be rate-limited to allow for the bandwidth boost to T_1. This expands on work by Singh Ott and Curcio and can be achieved using location information, sent by clients via the algorithm shown in Figure 4.9 (Sing et al, 2012) stored on the service' server, and querying a MNCM (*getUsersCentralInNearestSSA()*). Users falling into a nearby SSA classify as best-case users for use within the algorithm. The array of users is sorted in a descending fashion according to current service bandwidth of the users.

The server-side algorithm also defines how the stream service can determine the new rate-limit to be set on best-case users with "spare bandwidth" in the nearest SSA previously determined.

Calculating a suitable *maxRLC*(Max % rate-limit change) for the server-side algorithm can be calculated using a mathematical formula, shown in Figure 5.9, where *avgVideoBR* is the average video bit-rate of videos stored on the service, and *curGlobalRL* is the current global rate limit in place.

```
void bandwidthBoostReceived(){
remainingData = VideoSize – BufferedSize;
  time = distanceToNearestWSA(C)/GPS_getSpeed();
  possibleDL = currentBandwidth() * time;
if (remainingData >possibleDL)
requestLowerBitrate();
} else {
optimiseSystemForFastDownload();
}
}
```

Figure. 4.7 Server-Side algorithm to determine which users to rate (pseudocode)

Figure 4.8 Server-Side algorithm to determine which users to rate (Flow-Chart)

The result of the formula in Figure 4.9 is demonstrated to be 15Mb, although this may be lower in a real world scenario, depending on available server resources. Determining the best compromise between server costs and bandwidth availability for a given video stream service is not tested in this work.

```
maxRLC = rl * 100;
```

Figure 5.9 Mathematical formula to determine suitable maxRLC for Figure 5.5

The algorithm defined in Figure 5.10 is an alternative client-side algorithm, executed upon a successful bandwidth boost request. This algorithm is used to decide how to utilise the bandwidth increase. In cases where the full stream could be downloaded before entering the WSA, a fast download will be attempted – in other cases a lower bit-rate will be requested. Together, these algorithms and formulae proposed define a new stream service framework that aims to improve the QoS of a video stream to a user entering a WSA, without causing a widespread bandwidth deficit.

```
void handleBoostRequest(C, reqBW){
  maxRLC = 15;
  globalRL = globalRateLimit();
  minRL = globalRL – ((globalRL/100) * maxRLC);
  nActUsers = usersCentralInNearestSSA(C);
  optimumRL = reqBW/nActUsers;
  reqClient = getRequestingClient();
  boostedBW = 0;
  sortDescendingBandwidth(nActUsers);
  foreach(nActUsers as bestUser){
    curBW = getCurrentBandwidth(bestUser);
    spareBW = curBW – minRL;
    if (spareBW > 0){
      newRL = curBW – optimumRL;
      if (newRL < minRL){
        newRL = minRL;
      }
      boostedBW += curRL – new RL;
      throttleUser(reqClient, bestUser, newRL);
    }
    if(boostedBW >= reqBW){
      break;
    }
  }
  enableBandwidthBoost(reqClient, boostedBW);
  listenForCompletionOrDisconnect(reqClient);
}
```

Figure 4.10 Client-side algorithm to best utilize a granted bandwidth boost

5.2.2 Performance Testing

Two scenarios are used to test the performance of the proposed framework. Each scenario uses different values for each variable within the solution. Scenario 1 assumes a *maxRLC* (result of formula in figure 4.9) of 15Mb, whilst scenario 2 assumes a *maxRLC* of 20Mb. Each test scenario demonstrates how the proposed solution improves QoS for a user entering a WSA, in a different way.

5.2.2.1 Test Scenario 1

Figure 4.11 is a visual representation of the demo area used in this algorithm for performance testing of the proposed solution. Mathematical analysis will be used to analyse performance of the framework. Client or server resource performance/availability is out of the scope of this paper.

Using the algorithm from Figure 4.6, together with information from Figure 4.10 and Table 4.1, one can conclude that the required bandwidth per second required to complete the stream download is as follows:

$$reqBW = (30720 - 2048)/(60/1.5) \quad (4.5)$$

$$reqBW = 716.8\text{kb/s} \approx 5.6\text{Mbit/s} \quad (4.6)$$

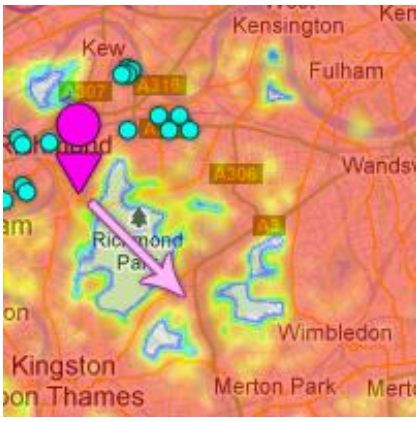

Figure 4.11 Demo area with T_1 (purple and B_k nearest best-case users *(blue)*)

Variables	Values
Current stream service rate-limit	4.0Mbit/s
T_1 speed (walking)	1.5m/s
Distance to nearest WSA (for *T_1*)	60m
T_1 heading toward WSA	True
Nearby active users in SSA	20
T_1 final video size	30Mb
T_1 buffered stream size	2Mb
Max rate-limit change	15%

Table 4.1 Variable Values for Test Scenario 1

User #	Current bandwidth
1	3.8Mbit/s ≈ 486.4kb/s
2	3.6Mbit/s ≈ 460.8kb/s
3	2.9Mbit/s ≈ 371.2kb/s
4	4.0Mbit/s ≈ 512kb/s
5	4.0Mbit/s ≈ 512kb/s
6	4.0Mbit/s ≈ 512kb/s
7	3.0Mbit/s ≈ 384kb/s
8	4.0Mbit/s ≈ 512kb/s
9	2.2Mbit/s ≈ 281.6kb/s
10	3.7Mbit/s ≈ 473.6kb/s
11	2.0Mbit/s ≈ 256kb/s
12	1.9Mbit/s ≈ 243.2kb/s
13	0.8Mbit/s ≈ 102.4kb/s
14	3.9Mbit/s ≈ 499.2kb/s
15	3.1Mbit/s ≈ 396.8kb/s
16	3.8Mbit/s ≈ 486.4kb/s
17	3.7Mbit/s ≈ 463.6kb/s
18	2.9Mbit/s ≈ 371.2kb/s
19	3.9Mbit/s ≈ 499.2kb/s
20	3.85Mbit/s ≈ 492.8kb/s

Table 4.2 Data for b_k best-case users illustrated in Figure 5.7

The algorithm from Figure 5.6 will send a request for a bandwidth boost since 5.6Mbit/s is greater than the current 4.0Mbit/s rate the client is currently receiving. This is an increase of 1.6Mbit/s, or 204.8kb/s. Continuing with the boost request server-side with algorithm 2 (Figure 4.8) and applying data from table 5.2 to Figure 5.6, the best-case users can be throttled appropriately as follows:

maxRLC = 15

minRL = 512 – (512*15%) = 435.2

optimumRL = 716.8/20 = 35.84

foreach nearest best-case user (Table II.):

newRL = *curBW* – 35.84

where: *curBW* 35.84 >*minRL*

The result of the algorithm defined in Figure 5.9. for test scenario 1 is displayed in Table 4.3. Any user with a lower bandwidth than the minimum determined by $minRL$ will default to $minRL$ – in this case 435.2kb/s ≈ 3.4Mbit. The total bandwidth that could potentially be saved, in this case, is greater than the extra bandwidth required by T_1. The server-side algorithm prevents unnecessary rate-limits being imposed and limits the best-case users with the highest bandwidth first. Users limited by this algorithm is highlighted in **bold**, *italicised* users were not required to achieve the requested bandwidth.

User #	New rate limitation
1	450.56kb/s ≈ 3.52Mbit/s
2	435.2kb/s ≈ 3.4Mbit/s
3	435.2kb/s ≈ 3.4Mbit/s
4	**476.16kb/s ≈ 3.72Mbit/s**
5	**476.16kb/s ≈ 3.72Mbit/s**
6	**476.16kb/s ≈ 3.72Mbit/s**
7	435.2kb/s ≈ 3.4Mbit/s
8	**476.16kb/s ≈ 3.72Mbit/s**
9	435.2kb/s ≈ 3.4Mbit/s
10	437.76kb/s ≈ 3.42Mbit/s
11	435.2kb/s ≈ 3.4Mbit/s
12	435.2kb/s ≈ 3.4Mbit/s
13	435.2kb/s ≈ 3.4Mbit/s
14	**463.36kb/s ≈ 3.62Mbit/s**
15	435.2kb/s ≈ 3.4Mbit/s
16	*486.4kb/s ≈ 3.8Mbit/s*
17	*463.6kb/s ≈ 3.7Mbit/s*
18	435.2kb/s ≈ 3.4Mbit/s
19	**463.36kb/s ≈ 3.62Mbit/s**
20	*3.85Mbit/s ≈ 492.8kb/s*

Table 4.3 Result of algorithm (Fig 5.9.) on $B_k$users (Table 5.2)

The new total bandwidth available to T_1 can be calculated as follows:

$$savedBW = \sum_{user=0}^{20} curBW_{user} - newRL_{user}$$

$$where: curBW > newRL$$

$savedBW$ = 215.04kb/s ≈ 1.68Mbit/s

Finally, using client-side algorithm 3 (see Figure 5.10), the client device can decide how to handle the extra bandwidth. The calculations for the algorithm will result as follows:

$remainingData$ = 30720 – 2048 = 28672

$time$ = (60/1.5) = 840

$possibleDL$ = (512 + 215.04) * 40 = 29081.6

Since 29,081.6kb is > 28,672kb, the device should optimise itself for a fast download – no bit-rate change is required.

4.2.2.2 Test Scenario 2

Steps from test scenario 1 are repeated with different values to represent another unique circumstance. Values for the test can be seen in Table 5.4.

Figures	Value
Current stream service rate-limit	3.0Mbit/s
T_1speed (car/public transport)	25m/s
Distance to nearest WSA for T_1	200m
T_1 heading toward WSA	True
Nearby active users in SSA	5
T_1 final video size	20MByte
T_1 buffered stream size	5MByte
Max rate-limit change	20%

Table 4.4 T_2, B_n Test Values for Test Scenario 2

User #	Current bandwidth
1	2.4Mbit/s ≈ 307.2kb/s
2	2.9Mbit/s ≈ 371.2kb/s
3	2.7Mbit/s ≈ 345.6kb/s
4	3.0Mbit/s ≈ 384kb/s
5	3.0Mbit/s ≈ 384kb/s

Table 4.5 Data for B_n Best-Case Users

Algorithm 1 calculations for test scenario 2:

$reqBW$ = (20480 – 5120) / (200/25)

$reqBW = 1920kb/s \approx 15Mbit/s$

Algorithm 2 calculations for test scenario 2:

maxRLC = 20

minRL = 384 – (384*20%) = 307.2

optimumRL = 1920/5 = 384

foreach nearest best-case user (Table 5.5.)

newRL = *curBW* – 384

where: *curBW* – 384 >*minRL*

Total saved bandwidth for test scenario 2

$$savedBW = \sum_{user=0}^{5} curBW_{user} - newRL_{user}$$

$$where: curBW > newRL$$

savedBW = 256kb/s ≈ 2Mbit/s

Result of algorithm 3 for test scenario 2:

remainingData = 20480 – 5120 = 15360

time = 200/25 = 8

currentBandwidth = 384 + 256 = 640

possibleDL = 640 * 8 = 5120

Thus for Test Scenario 2, the algorithm from Figure 5.6 would tell the client device to request a significantly lower bit-rate video stream, since only

5,120kb of the required 15,360kb can be downloaded with the increased bandwidth.

4.2.2.3 Test Scenario 3

Test scenario 3 shows the performance of the proposed framework against a service without any dynamic rate-limits in the same scenario.

Statistics for scenario 3 are detailed in Table 4.6, with a graph displaying results in Figure 4.12.

Figures	Value
Current stream service rate-limit	5.0Mbit/s
T_1speed (varied travel methods)	Various
Distance to nearest WSA for T_1	50m
T_1 heading toward WSA	True
Nearby active users in SSA	10
T_1 final video size	30MByte
T_1 buffered stream size	4.5MByte
Max rate-limit change	15%

Table 4.6 T_3, B_z Details for Test Scenario 3

4.2.3 Performance of framework (Scenario 3)

The performance of the proposed dynamic rate-limit is illustrated in Figure 5.12 showing a graph of the performance of the framework set out in Test Scenario 3 against a theoretical identical stream service with no dynamic rate-limits.

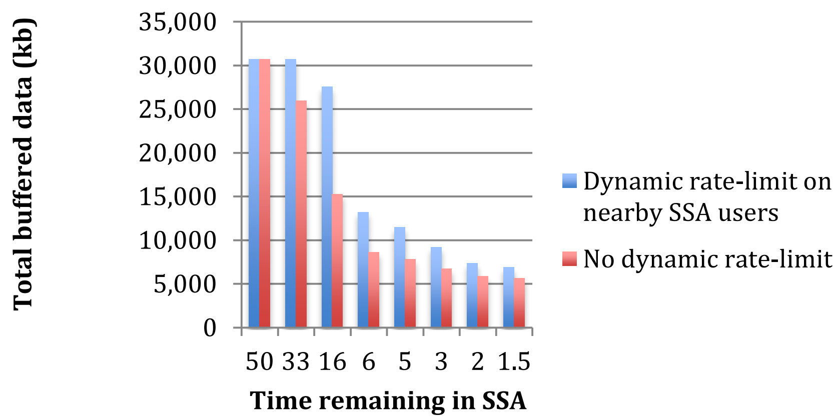

Figure 4.12 Performance of proposed dynamic rate-limit

4.2.4 Testing Analysis

In the three scenarios tested, the proposed framework vastly improves the QoS by maximising pause-less playback for a user predicted to enter a WSA. Scenario 1 shows how the framework can aid a user to complete their video at a consistent bit-rate, or quality. Scenario 2 demonstrates an example where a lower quality bit-rate must be requested – however, a bandwidth boost is still granted which will improve pause-less playback time if compared with a static rate-limit service in an identical scenario. Scenario 3 shows the performance improvement of a video stream service with the proposed framework implemented. A consistently larger buffer is downloaded with the framework in place, maximising pause-less playback. A new video stream service framework is proposed in this section, which uses dynamic rate limits to impose limitations on best-case users to improve the QoS for users predicted to enter a WSA. The location-based algorithms defined in this paper are new algorithms, mathematically tested with two different real world scenarios. In both scenarios, results showed an improvement in the QoS for the user predicted to enter a WSA. It is worth noting that the limitations of this framework is only suitable for popular video stream services with many users streaming at any given time, utilising most of the available stream service' bandwidth.

The proposal is resource intensive, in that it requires significant resources to prdict the signal strength, so perhaps currently best suited for specific cases. It does demonstrate that improvements in smoothing mobile network bandwidth fluctuations is possible, implementing the proposed video streaming service framework will improve the QoS for users entering a WSA.

4.3 Summary

This section proposes a video stream service framework that detects when mobile network bandwidth is likely to significantly drop for a given user; and defines dynamic rate-limitation algorithms to enhance the QoS of the video stream service for the given user without applying any global rate limits other users of the service.

The next chapter details the proposed Resource Harvest and redistribution CAC algorithm which builds on the pilot studies conducted in this chapter and further improves the QoS within the WiMAx domain. This is accomplished by utilizing redundant resources and redistributing them efficiently to new connection requests.

Chapter 5 Resource Harvest and Redistribution CAC Algorithm: Improving the Ratio of Successful Connections

This chapter investigates a method to harvest redundant resources such as bandwidth and QoS parameters, e.g. latency guarantees, which have been reserved for one connection but is not yet being utilised. As the WiMAX CAC the QoS parameter set work with the SF's to distribute resources to connections (section 2.10.2). This will then be redistributed to new connection request which would otherwise be rejected. The aim of harvesting resources is to improve the connection rates while utilising resources such as bandwidth more efficiently.

5.1 Resource Harvest and Redistribution CAC Algorithm

In the original CAC algorithm where the required QoS resources are not available within the provisioned parameter set, it has previously been ascertained that the connection request is automatically denied. To increase the number of active connections the reserved resources need to be redistributed to those connections that would have otherwise been denied, to reduce the number of rejected connections. To achieve this, first those connections that have been admitted but not having yet activated their allocated resources need to be identified (figure 5.1 (a)). Within this subset of connections the maximum (MSTR) and minimum (MRTR) resource requirements need to be identified. The MSTR parameter indicates the predicted required resources for that connection, but more importantly the MRTR parameter determines the level at which the connection can just maintain adequate QoS. Therefore in this instance the non-activated connections can be left with only the MRTR resources just in case they request to activate those resources in the future (figure 5.1 (b)). This would mean adequate QoS would be maintained for that connection, if activated, providing a MRTR, this method still provides adequate resources for the connection activation. Though only a proportion of these resources will be 'harvested' (figure 5.1 (c)), to ensure there is a buffer of Admitted resources available to protect the possible deterioration of QoS.

Subsequently, releasing the harvested resources to be redistributed to new connections, assuming the new connection is more likely to become activate (figure 5.1 (d)). The harvested resources will encompass those between the MRTR resource requirements and up to the MSTR resource requirements as stated. It is not guaranteed that the full complement of resources will be identified after searching through the non-active connections only. Though, it is essential that the full complements of requested resources are identified for any new connection, rather just than the minimum parameter requirement in the first instance. This part of the proposal is only harvesting resources from non-active connections. The aim is to improve the efficiency of resource allocation via the CAC policy and decrease the number of rejected connections.

After having undertaken this process if the levels of resources required for a new connection have not been identified from the non-active connections, and to prevent the new connection request being rejected at this stage, the second part of the algorithm needs to be invoked. Therefore the next step is to look at redundant resources allocated to the active connections while maintaining the actual MRTR needs for each current connection (figure 5.1 (e)).

This algorithm would identify those currently active connections that have a large proportion of their resources still available in the admitted but not yet activated QoSParamSet. Each connection requests resources based on assumptions of predicted use for that application type, the resources are then reserved in the AdmittedQoSParamSet these resources are not always fully utilized for the duration of the connection. Therefore this part of the algorithm focuses, only on those active connections that have surplus resources available in the admitted but not yet activated QoSParamSet. Consequently those resources that have not been utilized need to be ascertained. It is imperative that only redundant resources are identified and that the QoS of the active connection is not violated. As the connection can request additional resources from this parameter set, not all of it can be redistributed, as this will have a major impact on overall QoS.

The IETF RFC 2381 determined that a limit of 10% of resources could be harvested (IEFT, 1998). This is ascertained from the QoS requirements of real-time applications being able to tolerate a 10% decline of overall availability of resources before QoS deteriorates (Kappler, Fu, & Schloer, 2011). Even though this is 10% of the reserved resources only, it is assumed that if the connection requests all the remaining resources it will still be able to preserve the QoS of the connection, due to the fact that the level of resources harvested is within the stated range of the QoS tolerance. For example this would be 10% of those resources reserved but not currently activated, rather than 10% of those resources that was originally requested that is harvested from the non-activated connections. In the case of activated connections, taking 10% from the total resources requested has the potential for harvesting resources at such a rate that QoS could be compromised too quickly. Therefore just harvesting 10% from those resources that have not yet been requested will minimise the impact on the connection overall as there is still 90% of the AdmittedQoSParamSet resources remaining to enable additional requests for the connection to be granted.

The algorithm therefore harvests 10% of the reserved but currently unused resources per active connection to redistribute to the new connection request. In doing so the assumption is that if all the resources from the AdmittedQoSParamSet are then requested to be activated the resources will be comfortably in the bounds of providing adequate QoS. If at this point the level of required resources has not been achieved then the connection request will be rejected. But if the relevant resources have been identified they can then be redistributed to the requesting connection then the new connection is established and the SF will be assigned.

This enhancement will provide a more efficient use of those resources that otherwise may never be used. It increases the acceptance of more connections and prioritized those active connections to take full advance of those resources allocated to them. Additionally even if only small amounts of resources are harvested this would benefit the connection rate of the lower priority SF's without affecting the high priority request as these would have been rejected initially. The main disadvantage to this enhancement is the delay in searching for the required resources which impose additional overheads onto the system. The alternative is to reject the connection immediately therefore the cost of the short delay can result in an admitted connection that can utilize the harvested and redistributed resources

By monitoring the usage of the non-active connections potentially additional resources could be harvested to enable even more connections to be admitted, by removing additional portions of 10% of the admitted resources for active connections this process can iterate until 10% of the overall requested resources are harvested, after this point the connection may not be viable, if activation is requested (Kappler, Fu, & Schloer, 2011).

The harvesting of resources from current connections will only be completed for new connections and not just to replenish the ProvisionQoSParamSet. The ProvisionQoSParamSet will be replenished via connections that terminate or those that travel out of range of the BS so the communication quality can no longer be maintained.

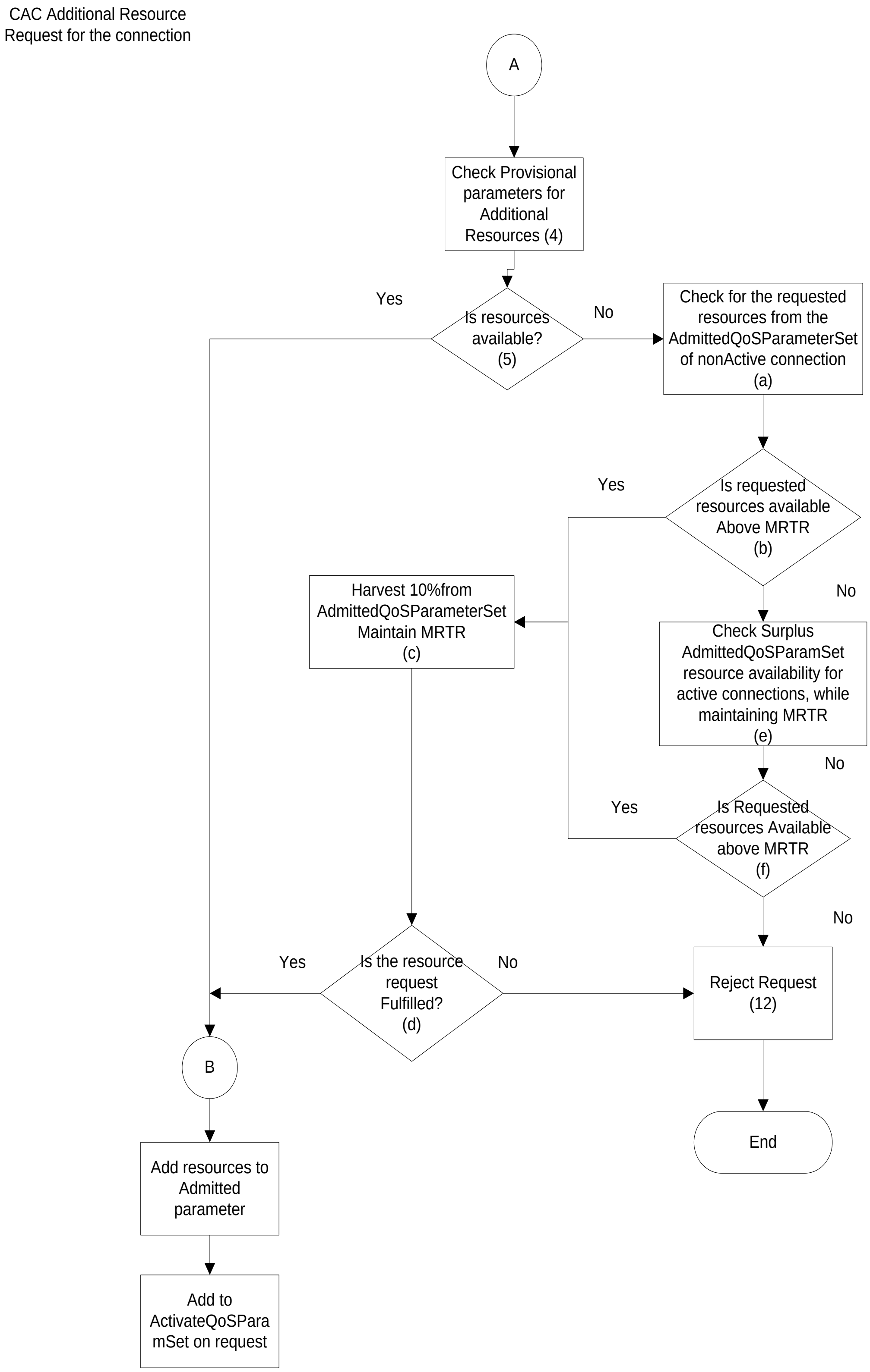

Figure 5.1 Resource Harvest and Redistribution – CAC Algorithm

Figure 5.1 outlines how the resource harvest and redistributed CAC (RHR-CAC) algorithm utilizes unused resources more efficiently to enable new connection requests to be granted. Initially the RHR-CAC algorithm follows that which is depicted in figure 4.4 until (A), then BS needs to ascertain whether the requested resources are available from the provisionedQoSParamSet (4, 5). If they are available then the CAC algorithm continues as intended (B). But if the requested resources are not available the connection initially would be rejected (12), but now the RHR-CAC algorithm would be invoked to prevent this from happening in the first instance, this does not guarantee that the required resources will be harvested but it does prevent the connection being instantly rejected when there is a possibility of the required resources being available. The worst case scenario is that there is additional delay before a final rejection.

Therefore the RHR CAC Algorithm will identify non active connections and then harvest 10% of their admittedQoSParamSet. On each harvest the total amount harvested is compared to the total amount required to determine if the harvest has fulfilled the requirements, this will iterate through all identified non active connections until all resources have been identified (a) . These will then be redistributed to the admittedQoSParamSet of the requesting SS connection (7). Depending on the relationship between the amount of resources requested and the quality of resources that are inactive, there may only be minimum latency injected into the process. The algorithm then reverts back to the original CAC process from (8), if all the requested resources have been identified.

If at this point all the resources have not been identified the RHR algorithm reverts to (e) where the connection that have surplus resources remaining in their admittedQoSParamSet are identified. The algorithm then iterates through this subset of connection to harvest 10% of the remaining resources again to the required quantity of resources have been achieved. Once these have been achieved the resources are then redistributed and added to the admittedQoSParamSet of the requested connection. Once the resources have been allocated the connection is then accepted. If the required resources have not been identified at this point the requesting connection is then finally rejected.

5.2 Bandwidth Redistribution UGS Simulation

A system that is both interoperable and supportive of QoS, while accommodating all SF's ensuring efficient bandwidth utilization is a challenging undertaking as was ascertained above in section 4.1. As stated in section 2.1 the SF UGS, which is used for real time applications such as VoIP, is being used, the BS periodically allocates a fixed amount of bandwidth to the SS. This can lead to bandwidth wastage when the packets the SS transmits fail to fill an uplink sub frame. This section discusses an enhancement to the UGS algorithm that will endeavor to improve the BU, by injecting efficiency into bandwidth allocation process (Peart, Adda, & Goodman, 2012).

By building intelligence into the algorithm, the aim is to reduce the amount of surplus allocated bandwidth. The BS would sample what has been transmitted so far on a periodic basis and compare this to the total amount of bandwidth that SS has actually allocated to it. If there is a significant difference, the bandwidth allocation can be reduced allowing surplus bandwidth to be reallocated to another SS. If it is found that the SS begins to use more of the bandwidth, more can then be reallocated as this is built into the standard.

$$B_T = \sum^{n}_{i=1} (B_{ai} - B_{ui}) + B_t \tag{5.1}$$

In equation (5.1) above n is the number of SS using the total available bandwidth, i is each instance of the connection, B_a represents bandwidth allocated to the SS, B_u is the bandwidth that is actually being used by the SS and B_t is the total bandwidth available. This calculates the surplus bandwidth by subtracting bandwidth from the allocated bandwidth and adds it to the total bandwidth available for redistribution to utilize a SS that requires it.

$$B_T = \sum^{n}_{i=1} (B_{ai} - B_{ui}) \leq B_{bi}, \tag{5.2}$$

Given a scenario where a SS transmitting real-time traffic that falls into the UGS class is sending packets through the uplink traffic that uses less than 80% of the allocated bandwidth (B_b) equation (5.2), under this algorithm the surplus bandwidth should be reassigned. Similarly more of the surplus bandwidth should be redistributed when utilization is close than 60%, 40% and 20%. Alternatively, if it is then found to be using close to 90% of the allocated bandwidth, the station should be allocated more bandwidth. As previously ascertained, real-time applications can tolerate a 10% reduction in bandwidth before QoS is impaired. The next section will detail the design of the simulation to establish the effectiveness of bandwidth allocation and utilization.

5.3 Bandwidth Allocation and Utilization Simulation

To simulate the proposed bandwidth redistribution enhancement a Visual Studio C# program was created. Within this model it is assumed that there are no interference, overheads or security issues that have been considered.

To model the data flow two timers were used, one to represent the packets being created by the user and another to represent the VoIP packets being successfully conveyed. The utilization of the bandwidth is calculated by measuring the data flow over a fixed interval, facilitated by a third timer. These timers call an attached process at selected intervals. Each time this third timer expires, a calculation is carried out to evaluate the average use of the bandwidth by the user. If the result of this calculation shows that less than 80% of the bandwidth is being utilized, the allocated bandwidth is reduced, and the third timer is restarted. If the transmitted packets increase in size, and more of the allocated bandwidth is then utilized to over 90%, more bandwidth is reserved. If the utilization percentage grows less than 5% the interval is slowed down to lower the allocated bandwidth. At each stage the average calculation and the amount of bandwidth allocated is recorded.

Based on a VoIP transmission using the G.711 codec, including overheads, packets of size 1744 bits will be generated at a rate of 50bps, (*Voice Over IP - Per Call Bandwidth Consumption*, 2006). Based on this information initial bandwidth allocation is set to 200Kbps to ensure complete accommodation of the resource requirements. Transmissions are averaged at a time interval of 200ms. Also included in this simulation are transmissions with packets of size 1104 and 624 bits.

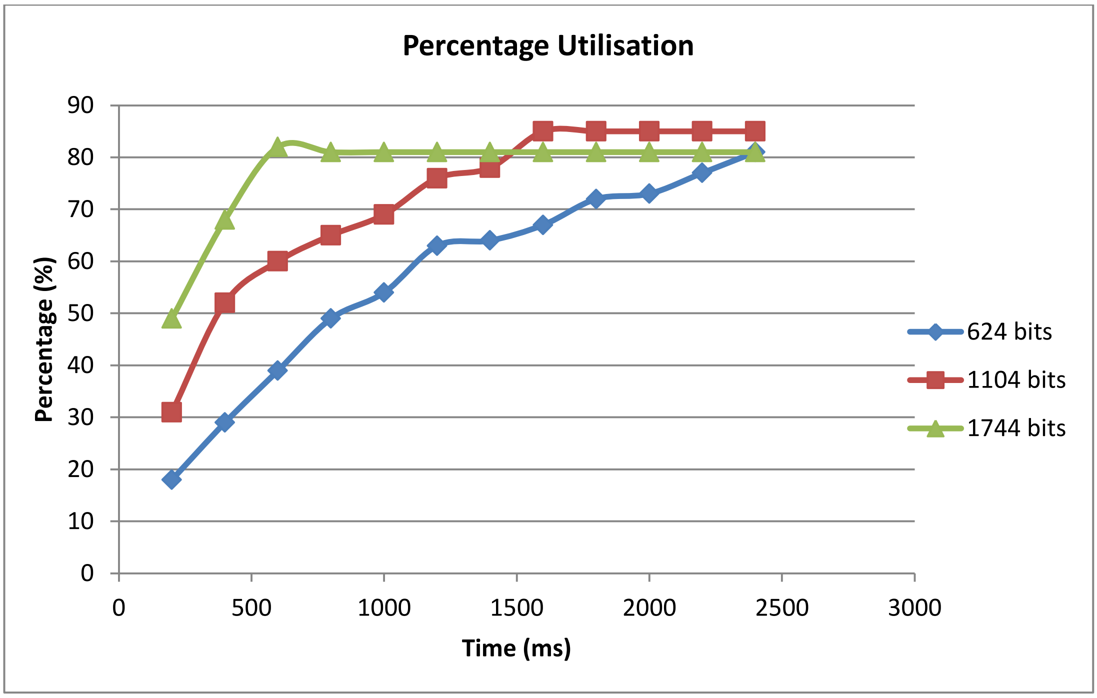

Figure 5.2 Bandwidth Utilisation

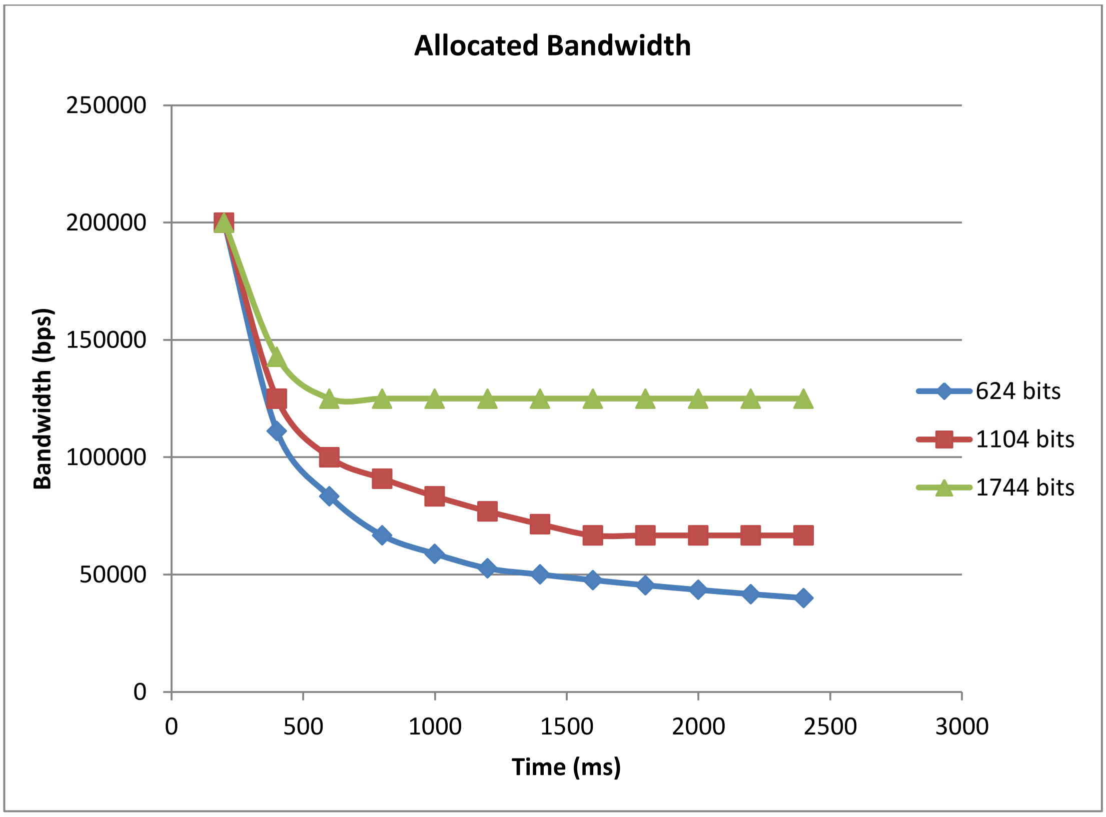

Figure 5.3 Allocated Bandwidth

Figure 5.2, illustrates that the bandwidth utilization increased over time, this coincides with the graph in Figure 6.3 which illustrates the allocated bandwidth decreasing over the same period of time. It can be seen that the transmissions including smaller sized packets that have less bandwidth allocated to them throughout the simulated time increased the bandwidth utilization which is evident in figure 5.3. For a constant transmission at this rate for ten minutes, this method would free up 45Mb of data capacity for reallocation when sending 1744 bit packets. The 1744 and 1104 bit bandwidth utilization hit similar optimum level at around 80% where it leveled out. The interesting outcome was that from the 624 bits transmission rate which steadily increased without leveling out within the duration of the simulation.

This simulation tested a bandwidth redistribution algorithm building on the current QoS SF within WiMAX aimed at improving bandwidth utilization for real time traffic with packets of a fixed size such as VoIP. This enhancement achieved a decrease of the allocated bandwidth until sufficient balance of resources is acheived. The algorithm has shown that specified amounts of reserved bandwidth were successfully removed until the bandwidth was being used to at least 80%. Overall a reasonable amount of bandwidth was redistributed for other requests. As in a WiMAX network, bandwidth is shared among many users; this improvement to the QoS will improve bandwidth efficiency sufficiently, allowing greater utilization of finite resources available

5.4 User Mobility Models in Ad hoc Networks

In today's technological world, users of mobile wireless devices are predominantly on the move while still enjoying connectivity of the Internet. How people use their mobile devices differ in many ways, not only from a technological point of view e.g. browsing the web, sending emails, SMS, downloading music/apps, or keeping up with friends on social network sites, etc. But also from a geographical point of view, the user's actual physical location, whether this is seated stationary in a park or shopping centre, where the user may only change location if the connection is poor, or actively mobile while interacting via the Internet, e.g. walking to work/college, while using VoIP/Skype or streaming media clips, (Briesemeister, Hartenstein, & Pérez-costa, 2004).

It is important to determine when simulating any QoS protocol, whether the perceived improvement will actually function under the planned usage, consequently it is vital to replicate the reality of the users behaviour (Camp, Boleng, & Davies, 2002; Davis, Eisenhardt, & Bingham, 2007). This is one reason researchers have investigated the characterisation of user behaviour in various network situations (Resta & Santi, 2006), such as the Dartmouth Campus (Jain, Lelescu, & Balakrishnan, Model T: and Empirical Model for User Registration pattrens in a Campus Wireless LAN, 2005; Kim & Kotz, 2005), here data collected from the IEEE 802.11 access points around the campus was analysed to determine actual user mobility. It has been identified that such specialised scenarios can limit the usefulness of the user model, and as a result the more general models can be an advantage (Resta & Santi, 2006). Consequently the environment in which the user inhabits can influence the way in which they will use their mobile devices, as in the Dartmouth Campus scenario. Therefore it is imperative to model generic user behaviour when simulating wireless protocols to represent the assumed reality of the actual mobility of the node's that will potentially utilize these protocols (Lee & Hou, 2006).

There are two core categories of mobility models, traces and synthetic models (Sanchez & Manzoni, Jan 1999). Traces once developed provide accurate information of the user's mobility, as they collate the actual mobility of the user that is observed over a period of time. The longer the period of observation coupled with a large number of participants produce precise data for example as in the Dartmouth Campus results (Lee & Hou, 2006). The disadvantage of this is that it is very intricate to model within a network environment (Law A. , Simulation Modelling & analysis, 2007). Also the trace has to be created before it can be tested. If the protocol has not yet been implemented, it is therefore impossible to first create the trace for testing a new protocol (Camp, Boleng, & Davies, 2002). Also traces can be constrained to the movement within the scenario they have been recorded, for instance the mobility on campus may differ to that recorded in a shopping centre. On the other hand the synthetic mobility model simulates the expected mobility of users, rather than emulates the actual movement of users as does the traces. The synthetic models still encompass changes in speed direction and in some cases the actual distance in ad hoc user behaviour. Kim, M; Kotz, D; Kim, S; (2006) validated their trace model by comparing synthetic traces with real traces, they concluded that synthetic traces match real traces with a median relative error of 17%, (Kim, Kotz, & Kim, 2006). Therefore inferring synthetic mobility models are adequate to model new protocols.

Therefore within the simulated environment it is essential to reflect the movement of the perceived user. In recognition of this several mobility models have been developed which encompass the Brownian Motion (Brown R. , 1828) and Random Walk (RW) (Johnson & Maltz, 1996), mobility models which also comprise models within transportation theory that have been incorporated in simulations of mobile networks (Briesemeister, Hartenstein, & Pérez-costa, 2004; Briesemeiter & Hommel, 2000; Helbing, 2001).

5.4 Random Walk (RW)

Natural movement is random and unpredictable. Consequentially the RW Mobility Model was developed to imitate this phenomenon, often referred as the drunkards walk (Resta & Santi, 2006). The model is designed to formalise the trajectory of the node moving from its current location to a new location by randomly selecting the speed and direction which are dictated by predefined ranges (Johnson & Maltz, 1996; Lee, Gerla, & Chiang, 1999). Fundamentally the RW paradigm is a mathematical model which formalises the trajectory of a moving node that is taking successive steps. It is strongly correlated to the diffusion models and is essential to the Markov processes. Once the node reaches the new location it pauses then a new speed $(s_{min}, s_{max})_i$ and direction $(0, 2\pi)_i$ parameters are calculated and the node begins the next phase of its journey j_i. This iterates within a given finite period of time and space. The RW tests the movement of nodes around a starting point, without them ever wandering outside the defined boundary. This was proven by Ploya in 1921 who ascertained that the node will return to the starting point with complete certainty (probability of 1.0), cited in (Weisstein, 1998; Daniel, Florian, & Wolfgang, 2011).

The 2-D variation of the RW model is synonymous to the 2-D representation of the Earth's surface. In this model the node commences its movement from the centre of the pre-defined area in the simulation, similar to the first model the node randomly selects the speed and direction to travel, the difference is that the node will travel a specified distance, rather than time before stopping and moving again. The RW Models is a memory-less mobility model, therefore no data is collected to record the previous movement of the node (Liang &

Haas, 1999; Hong, Gerla, Pei, & Chiang, 1999), nor is the current trajectory dependent on the previous node's movements, e.g. speed or direction (Hong, Gerla, Peri, & Chiang, 1999). The drawback of this model is that it can produce erratic behaviour such as abrupt stops and quick turns, which are said to be unrealistic in nature (Camp, Boleng, & Davies, 2002). Though if observing a typical teenager on their mobile device walking down the street the randomness of their moment emulates such erratic behaviour consisting of the sudden stops, see figure 6.5.

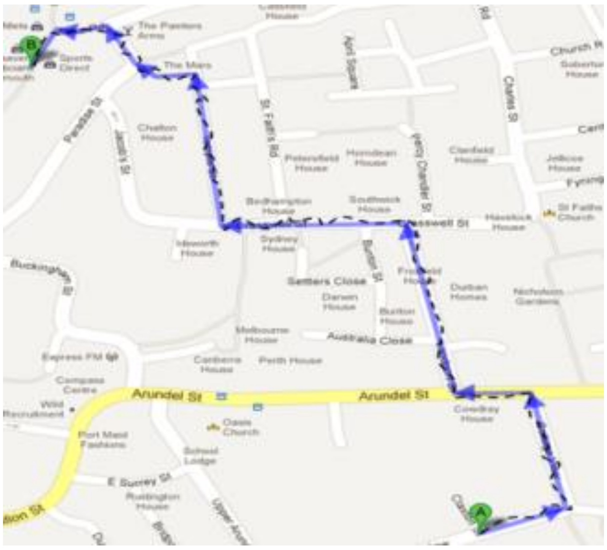

Figure 5.4 Route of a Typical Mobile Device User

Figures 5.4 to 5.6, illustrate the movement of multiple mobile nodes utilising the RW algorithm. To place the movement in context the background is to emulate a shopping centre scenario where the public often rely on such technology. These results derived from simulating the RW mobility model in NS3 (NS3, 2011). To accomplish this, the key parameters of the simulation are the area to be traversed, the default area shape is a rectangle, xy, the duration of the walk, t and the distance before recalculating the speed $(s_{min}, s_{max})_i$ and direction $(0, 2\pi)_i$. If the boundary is reached then a recalculation for the rebound angle and speed is invoked. For these simulations the speed is not a random parameter but has been converted to a constant speed to simplify the movement and isolate the randomness of the movement to the path traverse and eliminate the randomness of the speed in the initial stages of the experiments. For the RW mobility model, table 5.1 illustrates, the parameters used to generate the graphs, the number of mobile nodes recorded are 2,4,8,16,32. In figure 5.5 you can see two separate unrelated RWs from mobile nodes. Emulating the typical movement of shoppers using their mobile devices, here the pattern of the walk is clear.

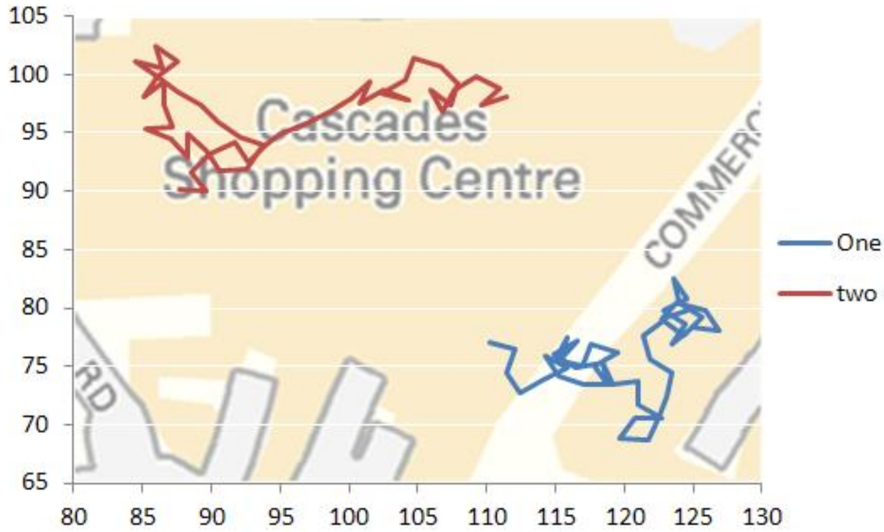

Figure 5.5. Random Walk Mobility Model Simulating Two Nodes

Parameter	**Value**
Time (t)	10
Speed (S_{min},S_{max})	Converted to a constant speed of 15
Direction(0,2π)	0.0, 0.628
Area (x,y)	(0|6000)(0|6000)
Starting Position	(0|100)(0|100)
Total Simulation time	100
Mode	Time

Table 5.1 Parameters used in the simulations for the Random Walk mobility model

In figure 5.6 and figure 5.7 one can see as the number of mobile node traces increase there are more intersections of the paths which could result in fiercer competition for a stable connection. By the time 16 mobile node paths have been recorded (figure 5.7) differentiating between node paths become more difficult. When 32 and above mobile nodes are recorded (figure 6.8) the paths become almost impossible to differentiate, if this were simultaneous mobile nodes paths in a real-time situation the simulation could be more suited to a large crowded scenario such as a music festival or a busy theme park.

Figure 5.6. Random Walk Mobility Model Simulating Sixteen Nodes

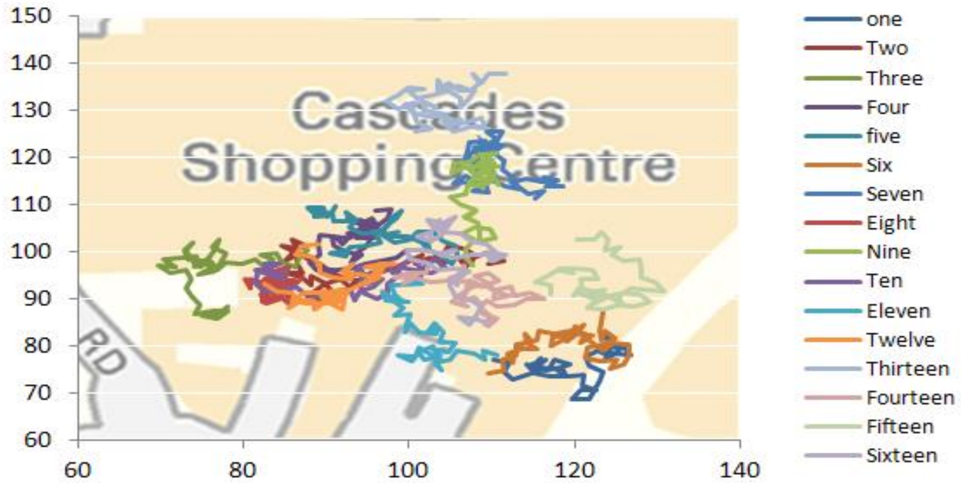

Figure 5.7 Random Walk Mobility Model Simulating Thirty Two Nodes

5.5 Random Waypoint

Random waypoint (RWP) mobility model is a common mobility model that is used in the simulation of adhoc wireless networks (Rubin & Choi, 1997; Bar-Noy, Kessler, & Sidi, 1994; Zonoozi & Dassanayake, 1997). In the RWP model, the node initially identifies a random destination which is referred to as the ‘waypoint’, the actual movement is at a random speed, emulating the RW model (s_{min}, s_{max}). The process commences from a random point in a given space $[x_i, xy_i]$, with the node first pausing $[p_i\ ]$, and then moving toward the pre-defined destination, Figure 6.9 illustrated expected node movement. Once the node reaches its destination it again pauses before moving to the next pre-defined destination, then the process iterates until it is finally terminated. The key difference between this and the RW model is that it introduces a pause time once the node reaches the pre-defined destination (Johnson & Maltz, 1996). Therefore by setting the pause time to zero the model would emulate the RW model (Camp, Boleng, & Davies, 2002).

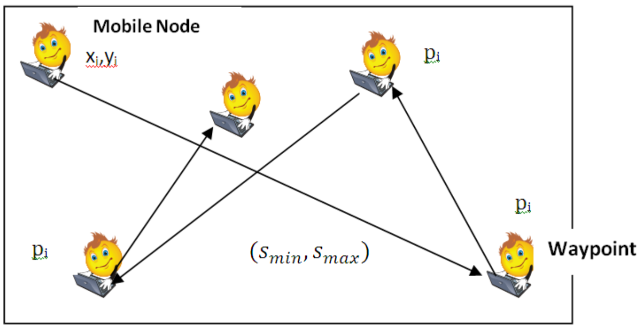

Figure 5.8 Typical Movement Produce for the Random Waypoint

Camp, Boleng, Davies, (2002), concluded that this model can produce a variable average neighbour percentage for the first 600 seconds of the simulation resulting in a high variability of the simulation results. If this is an issue for the experiment Camp et al, suggest discarding the first 1000 seconds of the simulation to produce an initial configuration period (Camp, Boleng, & Davies, 2002). To establish if this occurs within the experiments conducted, multiple runs of each simulation will determine if there is any variability in the result produced. Karp and Kung (2000) states that longer pause times will produce a more stable network even at high speeds, as overall this emulates a more static network (Karp & Kung, 2000). The paths of movement produced for both the RW and the RWP simulations are similar, due to the pause parameter not being visual within this graphical type. Therefore only a single node has been illustrated in figure 5.9, utilizing the same scenario as for the RW model.

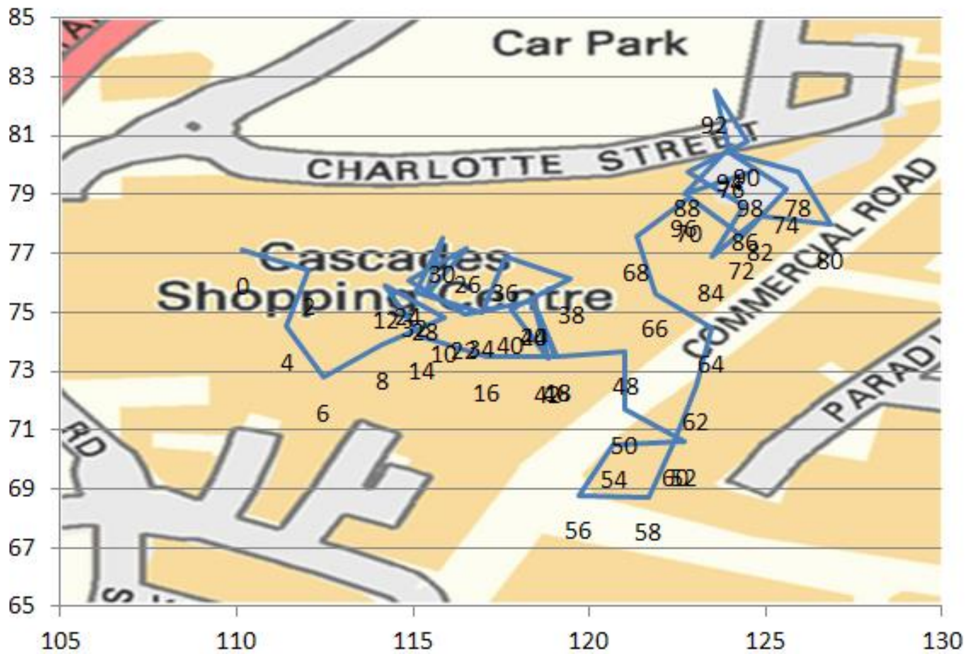

Figure 5.9 Single Node Mobility using the Random Waypoint Model

5.6 Gauss-Markov

Gauss-Markov was originally developed for a Personal Communication Service (Liang & Haas, 1999), and more recently incorporated in simulations of ad hoc networks (Camp, Boleng, & Davies, 2002). The Gauss-Markov mobility model uses one parameter to fine tune the levels of randomness of the model from a Gaussian distribution. Each node is allocated its speed and direction which is updated periodically. At each point in time the speed and distance is calculated based on the previous trajectory of speed and direction variables. Any node that lingers by the boundary of the simulated area are actively moved if they are within a pre-defined distance of the boundary.

The Gauss-Markov model reduces the concept of unexpected turns that is fundamental to both the RW and the RWP models, therefore said to emulate a more realistic model of mobility. But as previously acknowledged everyday movement often incorporates unexpected stops, especially if the user is walking while using their mobile device or travelling in a vehicle within a congested city or motorway. The Gauss-Markov model consumes more computational power due to the number of parameters needed to determine the motion of the next step based on the previous position to maintain constant motion. To eliminate abrupt pauses and sudden turns, data needs to be collated from the last move variables to ascertain the future trajectory variables. In equation (5.1), (Camp, Boleng, & Davies, 2002) one can see how the velocity of the node's trajectory is determined over time. The mean value of v_n is represented by the constant μ. α is the random tuning parameter 0 ≤ α ≥ 1, setting α=0 to represent Brownian motion and α=1 to represent the linear motion. x_{n-1}.

$$v_n = \alpha v s_{n-1} + (1-\alpha)\mu\alpha\sqrt{1-\alpha^2}\, x_{n-1} \qquad (5.1)$$

Figure 5.10 Illustrated a trace of one node moving within the boundary of a 300, 600 (x, y) rectangle that utilizes the Gauss Markov mobility model. As shown, the movement is more fluid with no abrupt stops coupled with sudden turns as evidenced in the other models. To generate the trace the speed and distance changed every second; the next direction and speed of the node is determined via a random Gauss variable generated from 1 to 6.28. The use of one node for figure 8 is selected to give clarity to the movement of the model, a trace of multiple nodes distracted from the fluidity of the movement and the elimination of sudden turns becomes unclear.

Figure 5.10 A One Node Trace of a Mobile Node using the Gauss-Markov Mobility Model.

5.7 Environmental Effects On Mobility

This proposed mobility model that has the intention to enhance current synthetic model interaction of nodes with their environment including social factors of movement. This environmental model (EM) will highlight the interaction and correlation between nodes and real world movement. Obstacles are included to replicate real environmental effects on mobility and used as waypoint destinations for nodes, emulating a campus scenario incorporating the realistic movement of MNs using different paths to move around the campus buildings while travelling to their next lecture hall. Figure 6.12 illustrates the possible angle increments for MN movement, while figure 10 demonstrates the execution of the proposed model when a node (N) is faced with an obstacle impeding its path to its goal (G). When encountering an obstacle realistically humans would not bounce off a building, in the EM nodes repeatedly increment their traversal angle to the nearest 45 degree until they're able to successfully move around an obstacle as shown in figure 5.11 with the node increasing from 150^0 to 225^0. Upon reaching a goal the node pauses before selecting its next destination see table 5.2. Similar to the Random Waypoint (RWP) this allows the control of movement of MN, but in this model it also effectively controls the proximity of the nodes to obstacles, the pause parameter can be used to represent the student pause times at buildings as they attend lectures.

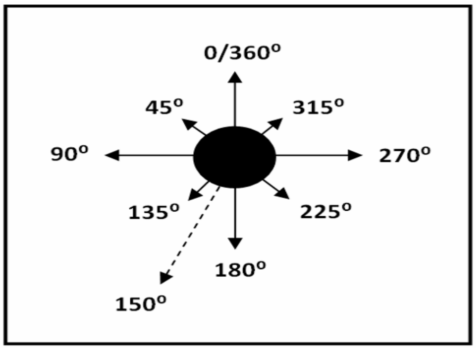

Figure 5.11 The possible angle increments for movement

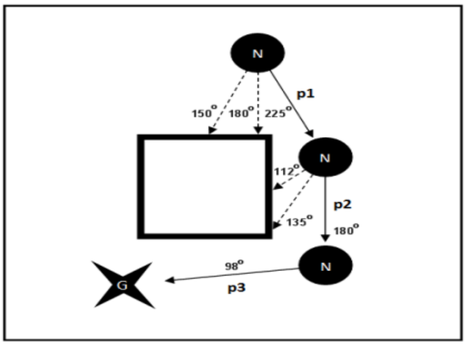

Figure 5.12. Implementation of the proposed model when a node (N) is faced with an obstacle impeding its desired path to its goal (G)

	1	2	3	4	5
1		*0.3*	*0.7*	*0.9*	*0.1*
2	*0.3*		*0.4*	*0.2*	*0.8*
3	*0.7*	*0.4*		*0.6*	*0.1*
4	*0.9*	*0.2*	*0.6*		*0.4*
5	*0.1*	*0.8*	*0.1*	*0.4*	

Table 5.2: An Example of an Adjacency Relationship Matrix between 5 nodes

5.8 Social Theory Effects On Mobility

Based on the same ideology from social network theory weighted values are assigned to represent the relationships between each node, through an adjacency matrix, with 0.1 being a weak relationship and 0.9 being a strong relationship. These relationships are used for various interactions and affecters of node movement in the model. Interactions between the nodes take the form of 'conversations', which can be initiated when two nodes are in close proximity. The chances of the two nodes pausing is based on the strength of their relationship, a relationship of 0.7 means the nodes will initiate a conversation 7 out of 10 times. Two nodes will pause their traversals for a conversation of a set amount of time again based on the strength of the relationship, the stronger relationship the longer the pause and store the previous node's ID to prevent repeat conversations. From these conversations groups can form, similar to that of social theory, as more nodes will join the conversation.

The node's weighted relationships are used to further integrate social impact by having a node's strong relationships influence its next destination. This is implemented by taking the current destinations of nodes with a high relationship value and adding these current destinations to the next random destination choice of the node thereby biasing the choices. A node is then more likely to choose a destination where it can encounter its friend's thus additional grouping or communities can form. Further to this with each traversal a random number is generated to cause a direct impact on the node's movement by increasing or decreasing its movement angle to a slighter degree. The inclusion of this angle modifier results in a less synthesized human movement than previous models have where nodes traverse in straight lines towards their goal.

5.9 Environmental Model

The environment is populated with a number of obstacles (*obs),* which are assigned entrances as a list of possible random destinations for all nodes (n), added randomly. Nodes have a relationship with other nodes through an adjacency matrix. Each interval nodes move towards their destinations by calculating the required angle and adding a random angle modifier, by incrementing its angle by 45 until it's possible to move. If another node is in proximity and relationship value is strong then initiate conversation and the pause traversal is invoked. When a node reaches its destination it will pause (*pp).* Then a new destination is selected randomly from a list of potential destinations plus the current destinations of the node's high weighted relationships. The EM and RWP are compared to ascertain the effects on the node mobility. To analyse the models in Repast both scenarios contained 5 nodes with the same set node velocity. Node density was analysed by noting the proximity of the nodes at 30 seconds intervals see figure 5.14 and the final value is an average of these intervals see figure 5.13. It can be observed in figure 5.15 that there is a dramatic increase in the overall closeness of nodes in the EM when compared to the RWP model showing an increased correlation between the interaction of nodes and the overall closeness of nodes. In addition that by increasing the number of obstacles node interactions can be disrupts and their mobility patterns altered. Finally, figure 5.15 also shows the instability of the node density in the model, pointing to the formation of groups of nodes at random periods within the simulation.

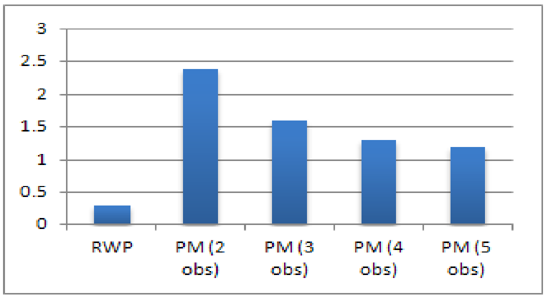

Figure 5.13 The average node densities of RWP and the proposed model with a varying number of obstacles

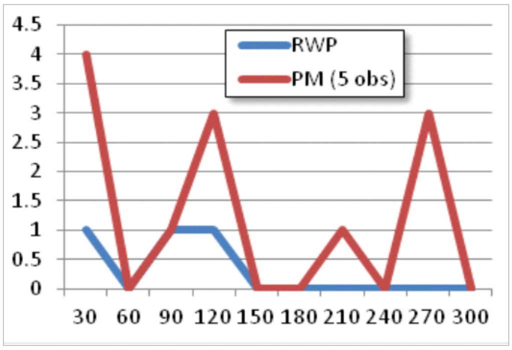

Figure 5.14 The node density values for RWP and the proposed model with 5 obstacles.

5.10 RHR-CAC Analysis

Mobility is an essential part of using modern devices, the random walk model has been identified as the one that closely relates to the average user without having collated knowledge of the actual data from physical movement of a the actual user. It was also established even with this data that simulations would become complex coupled with the issue that certain patterns of movement are particular to certain environments, such as the Dartford campus and are not interoperable with everyday use. Therefore this work uses the random walk mobility model. The following data is based on 32 nodes utilizing all the SF categories (5.18) with a variety of QoS requirements. The bandwidth with be using QAM 64 with 42mbps of throughput (figure 5.16 & 5.17).

BW [MHz]	n	F_S [MHz]	N_{FFT}	Δf [kHz]	N_{data}	T_b [µs]	T_s [µs]	R_{QPSK} [Mbps]	R_{16QAM} [Mbps]	R_{64QAM} [Mbps]
10	28/25	11.2	1024	10.9	720	91.4	102.9	14	28	42
8.75	8/7	10	1024	9.8	720	102.4	115.2	12.5	25	37.5
7	8/7	8	1024	7.8	720	128	144	10	20	30
5	28/25	5.6	512	10.9	360	91.4	102.9	7	14	21
3.5	8/7	4	512	7.8	360	128	144	5	10	15

Figure 5.15 Maximum Instantaneous Gross Data Rates (Rohde & Schwarz, 2014)

FEC coding rate	Max DL payload data rate [Mbps]			Max UL payload data rate [Mbps]		
	QPSK	16QAM	64QAM	QPSK	16QAM	64QAM
1/2	3.168	6.336	9.504	2.016	4.032	6.048
2/3	n/a	n/a	12.672	n/a	n/a	8.064
3/4	4.752	9.504	14.256	3.024	6.048	9.072
5/6	n/a	n/a	15.840	n/a	n/a	10.080

Figure 5.16 Maximum payloads gross data rates BW 10MHz, PUSC G1/8 (Rohde & Schwarz, 2014)

Type	rtps	nrtps	BE
DVD	2		
YouTube – 400kbps	2		
YouTube – 1Mbps	1		
YouTube – 2.5Mbps	3		
MPEG2	2		
MPEG4	2		
Internet browsing		5	
Graphics		5	

FTP			4
SMTP			6

Table 5.3 The Application and SF Type for Connections

Table 5.3 compares the number of connections granted based on the two scheduling algorithms, firstly via FIFO and secondly via WFQ. This tested the 32 possible connections, which all attempted to be activated, with surplus bandwidth available in the admmitedQoSParamSet for the granted connections. Each SF categories have been considered. Initially no resources were harvested or redistributed to those connections that had been denied a connection. The FIFO implementation illustrated that the higher priority SF's were allocated the data first and the lower SF's were starved of a connection, which confirms the literature in section 3. This also meant that the available bandwidth was allocated to the first connection requests which resulted in fewer connections being admitted. Utilizing this scheduling algorithm as part of the CAC, provides no QoS support and only based on the order that the connection requests are received. The WFQ granted connections and balanced the available bandwidth with the QoS requirements of each connect. The queue of requests had to be investigated to apply the weighting to provide QoS coupled with a better rate of connections in comparisons to FIFO.

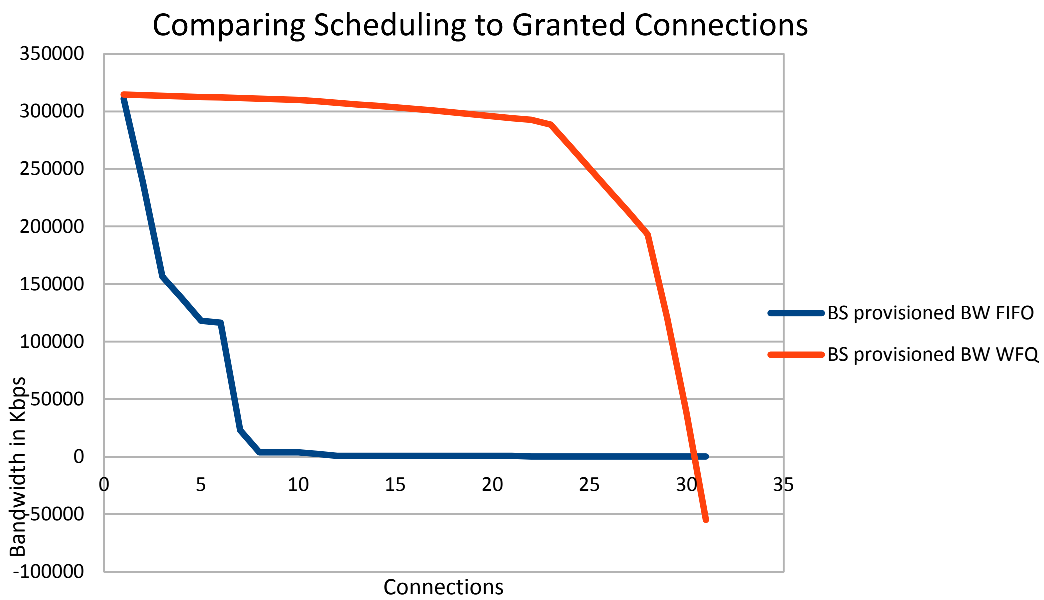

Figure 5.17 FIFO and WFQ Scheduling Comparison for Granted Connections

Harvesting resources from those connections that have firstly not activated their connection and secondly from the admittedQoSParamSet variable for those not utilizing their full complement of resources. As discussed earlier, 10% of resource can be harvested to provide additional resources for new connections of those requests that would normally be denied. Also based on the earlier work, (see section 6.2), that focused on resource distribution for WiMAX UGS which identified a successful method to include additional resources to grant connections.

Based on the data used for each connection in using all SF's 3.6Mbps was harvested for connections that were activated therefore have a small amount of surplus bandwidth available in the AdmittedQoSParamSet which was not being used. For the CAC using WFQ the harvested resources were insufficient to grant additional connections in this scenario, this was due to the efficient allocation of resources based on the weighting. This resulted in the connections being denied initially not being able to gain enough resources via the harvesting algorithm either, given the same bandwidth utilisation and efficiency. But in the case of the FIFO application of there was improvements of granted connections, this was due to the lower priority SF being starve of a connection. Here the surplus resources enabled these to gain connections that previously would be denied (figue 5.18).

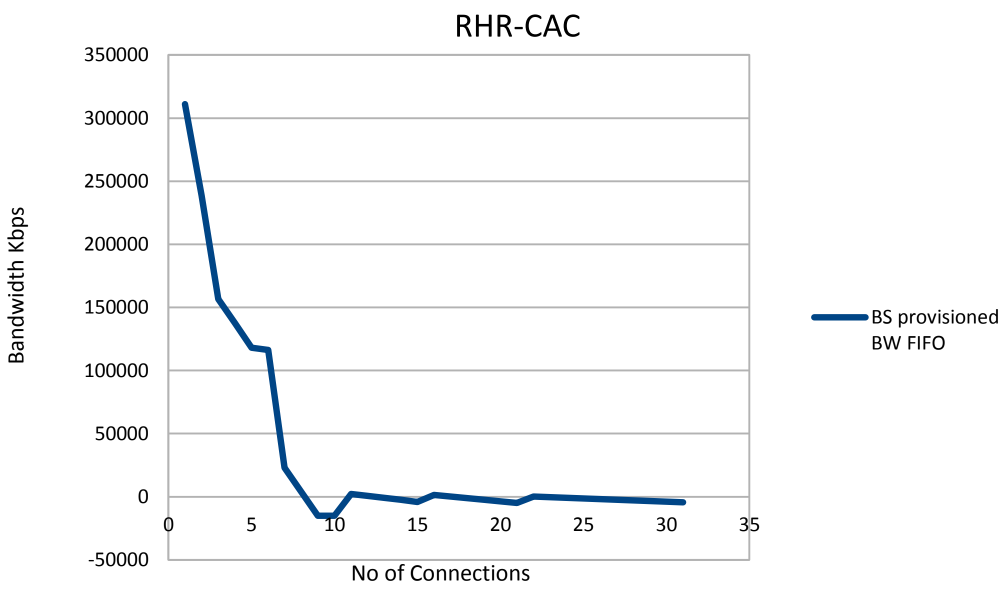

Figure 5.18 FIFO scheduling after RHR-CAC

5.11 Summary

These models are not without their faults, the RW model requires the spatial distribution of the node which is non-uniform (Bettstetter, Resta, & Santi, 2003). Also there is decay in the speed when nodes are undertaking extended periods of mobility (Yoon, Liu, & Noble, 2003). As generally mobile device users do not walk in a uniform fashion with equal spatial distribution nor at a uniform speed.

The RW Mobility model that utilizes small parameters for the distance or the duration of the movement will produce a Brownian motion with minimal movement. Substantially increasing the parameter replicates a RWP without a pause. This model has been used in many prominent simulation studies of ad hoc network protocols (Camp, Boleng, & Davies, 2002; Hong, Gerla, Pei, & Chiang, 1999; Lee & Hou, 2006). The model has some flexibility built into it and can replicate a certain level of reality of node movement. Though the key issue is that it assumes all movement is in a straight line pattern between destinations. The Gauss-Markov model also has the ability to replicate real-world patterns of movement though the parameters need to be carefully selected to produce this.

Camp et al (2002) recommend the use of either the RWP or RW mobility models if an entity mobility model is required. Key indicator to this decision is the availability of the models within the simulation packages and that with a few shortcomings mobility of the real-world to an expectable level is replicated (Camp, Boleng, & Davies, 2002). The results shown in figure 3 to 8 agree with camp et al (2002), recommendation, both models can emulate movement which typifies mobile users in an ad hoc network. The EM overcomes this limitation by encompassing social and environmental factors into the simulation. The constraint of this model is that social connection and destination needs to be known before commencing the simulation and it too adds more computational requirements to the process.

Chapter 6 Conclusions

This chapter presents the summary of the research of this thesis, the limitations of the work and describes the potential for future direction. The main motivation for this work was the basic human need to communicate and development of communication methods, which has rapidly evolved in recent years. Inferred from Mongtagu's work, communication became the foundation of community. '*It is, in short the essential human connection*' (Montagu, 1979). The quicker and wider reaching the communication is, the faster humans are able to learn and develop.

John Ruskin (1819-1900) statement of "*Quality is never an accident; it is always the result of intelligent effort.*" Though this is true, this work illustrates it is more than that, within the domain of network communication. It is the intelligent trade-off between key variables, to finely balance '*quality*' to enhance the end users experience. Quality is synonymous with excellence but is excellence actually the requirement? QoS has cost and time constraints, which needs balancing to provide the end user reliable functionality at the point of need.

The focus of this work was to ascertain whether '*Quality of Service can adequately be maintained for real-time streaming multimedia applications in an ad hoc wireless network environment*' particularly within the WiMAX domain. To establish this two pilot studies were undertaken the first utilizing the iABC. The iABC provides an interesting proposition to provide all parties with their respective requests, while maintaining a high and maintainable QoS for all. It did however, come at a cost in terms of delay to the required service request. But as in most things with communications networks, that there is always a trade-off. Is the additional delay worth getting a connection and bandwidth over an instance denial? The second was predicting mobile network bandwidth fluctuation to enhance video stream service quality, where three scenarios were developed for testing.

The proposed solution improved the QoS by maximising pause-less playback for a user predicted to enter a WSA from a SSA. The first scenario substantiated a user can complete their video at a consistent bit-rate, while maintaining adequate quality. The second scenario evidenced, when a lower quality bit-rate is requested, a bandwidth boost supports pause-less playback compared with a static rate-limit service in an identical scenario, which provided a degraded QoS. The third scenario utilized the algorithm resulting in a consistently larger buffer which in turn maximised pause-less playback. The results illustrated that the video stream service framework, improved QoS by using dynamic data rate limits to impose limitations on best-case users predicted to enter a WSA. The limitation is this is only adequate for popular video streaming services with many users streaming at any given time, where the bandwidth is at saturation point.

The proposed framework is resource intensive and perhaps currently best suited for specific cases. It has shown that an improvement in smoothing network bandwidth fluctuations is possible. By implementing the proposed video streaming service framework will improve the QoS for users entering a WSA.

In achieving an optimal system performance with an appropriate amount of fairness throughout the PHY and MAC layers. Cao and Li (2001) identifies five key issues in wireless networking, the variability of the actual wireless link; fairness of the throughput of the transmission; QoS to guarantee the required parameters are available for the transmission whatever the condition of the network; data throughput and channel utilization ensuring efficient use of the available network resources and power capacity (Cao & Li, 2001).

The iterative approach of the research methodology has been beneficial as it has provided the mechanism for smaller experiments to test the viability of QoS in a variety of wireless domains before focusing on WiMAX. Aspects of this algorithm are interoperable with other wireless environments especially when scheduling for traffic priorities.

7.2 Limitations of the research

This work has taken an incremental approach to challenging the QoS within wireless networks. Starting From increasing connection rates via increasing buffer sizes to allow additional connections, then providing a framework to enable the prediction of users transferring from a SSA to a WSA and smooth the transition to maintain the QoS of streaming media.

From here the work with the RHR-CAC evidenced that the bandwidth can be evenly harvested and redistributed but the trade-off is delay. This algorithm is only required if the BS resources cannot fulfil a connection request, therefore the balance is a latency to possibly gain a connection rather than an instant denial of a connection.

In all experiments, neither channel interference nor security aspects of the system has been considered. This could be considered as future work.

7.3 Future Work

The future direction for predicting bandwidth fluctuation can be to consider multiple users in a given area predicted to enter a WSA at a similar time, and how to calculate more appropriate rate-limits for such cases. One possible solution for this would be to look at other nearby SSAs for more best-case users in order to further disperse the required rate-limitation.

Future work within the RHR-CAC algorithm would be to incorporate multiple BS within a mesh topology. This could include more efficient handoff's between BS's to further improve efficiencies of bandwidth utilisation. This would consider smoothing the load of overloaded BS's to those that are less loaded. The ranging and connectivity of the SSs from the BS would need to be considered.

The ultization of the RHR-CAC algorithm would enhance bandwidth constrained areas such as rural bandwidth constrained area of Britian. This would ensure that efficient use of the network resources are used optimally. Future work would encompass environment disaster areas to support rescue/humanitian aid workered in areas where network connectivity is challenging.

References

Abbott, D., Davies, B., Phillips, N., & Eshrahian, K. (1996, Feb). Simple Divation of the thermal noise formula using window-limited fourier transforms and other conundrums. *IEEETransactions on Education, Vol 39*(No1), 1-13.

Abramson, N. (2009, December). "The ALOHAnet – Surfing for Wireless Data. *IEEE Communications Magazine, 47*(12), pp. 21-25.

Adda, M., & Peart , A. (2006). High priority traffic in HCF on wireless networks. *Telecommunications and Computer Networks, IADAT-tcn* (pp. pp 67-72). Portsmouth: IADAT.

Adda, M., Peart, A., & Watkins, N. (2006). Quality of Service in Wireless ATM for High Demand Multimedia Applications. *2nd Information and Communication Technologies, 2006* (pp. pp 3233 - 3238). New York: IEEE Xplore.

Adler, R. B., & Rodman, G. (2006). *Understanding Human Communication* (9th ed.). New York: oxford university Press.

Aho, A. V., & Ullman, J. D. (1992). *Foundations of Computer Science.* New York: W. H. Freeman.

Allan, D. (2014, April 24). *Does your street feature the slowest broadband speeds in the UK?* Retrieved from ITproportal: http://www.itproportal.com/2014/04/24/slowest-broadband-streets-in-the-uk-named/

Anderlind, E., & Jens, Z. (1997). A Traffic Model for Non-Real-Time Data. *IEEE COMMUNICATIONS LETTERS, VOL. 1*(NO. 2), pp 37-39.

Andrews, J. G., Ghosh, A., & Muhamed, R. M. (2007). *Fundamentals of WiMAX.* Massachusetts: Pearson Education Inc.

Andrews, J., Ghosh, A., & Muhamed, R. (2007). *Fundamentals of WiMAX: understanding Broadband Wireless Networking.* Massachusetts: Prentice Hall.

Andrews, M., Kumaran, K., & Ramanan, K. (2001, February). Providing Quality of Service over a Shared Wireless Link. *IEEE Communications Magazine*.

Aniba, G., & Aıssa, S. (2004). Adaptive Proportional Fairness for Packet. *IEEE Communications Society Globecom*, pp4033-4037.

Atterbury, P. (2011, 2 17). *Victorian Technology.* Retrieved from BBC History: http://www.bbc.co.uk/history/british/victorians/victorian_technology_01.shtml

Avalan. (2014, Decemeber 29). *Avalan Wireless Blog*. Retrieved from Avalan Wireless : (http://info.avalanwireless.com/blog/topic/900-mhz)

Banks, J., Carson, J. S., Nelson, B. L., & Nicol, D. M. (2005). *Discrete-Event System Simulation* (4th ed.). Upper Saddle River , New Jersery: Pentice-Hall.

Bar-Noy, A., Kessler, I., & Sidi, M. (1994). Mobile Users To Update or Not to Update? *Computer Communications (INFOCOM'94)*, (pp. 570-576).

Beavis, G. (2013, feb 20). *techradar*. Retrieved August 31, 2013, from What the 4G auction means for you: which networks won and lost?: http://www.techradar.com/news/phone-and-communications/mobile-phones/what-the-4g-auction-means-for-you-which-networks-won-and-lost-1132571

Belrose, J. S. (1995). Fessenden and Marconi: Their Differing Technologies and Transatlantic Experiments During the First Decade of this Century. *International Conference on 100 Years of Radio, 5-7 September* (pp. 10-19). Canada: Communications Research Centre.

Bettstetter, C., Resta, G., & Santi, P. (2003, July-Sept). The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks. *IEEE Trans. On Mobile Computing, 2*(3), 257-269.

Bio. True Story. (2013). *Alexander Graham Bell biography*. Retrieved Nov 2013, from Bio. True Story: http://www.biography.com/people/alexander-graham-bell-9205497

Bletchley Park. (2013). *Pigeon at War*. Retrieved Dec 27, 2013, from Bletchley Park: http://www.bletchleypark.org.uk/content/visit/whattosee/pigions.rhtm

Borge-Holthoefer, J., Baños, R. A., & Moreno, Y. (2013). Cascading behaviour in complex socio-technical networks. *Journal of Complex Networks*, 1, 3–24.

Briesemeister, C., Hartenstein, H., & Pérez-costa, X. (2004). Stochastic Properties of the random Waypoint Mobility Model. *Wireless Networks, 10*(5), 555-567.

Briesemeiter, L., & Hommel, G. (2000). Role-based multicast in highly mobile but sparsely connected ad hoc networks. *Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc).* Boston, MA.

British Telephones. (2010, December 20). *UK TELEPHONE HISTORY*. Retrieved Decemebr 27, 2013, from UK TELEPHONE HISTORY: http://www.britishtelephones.com/histuk.htm

Brown, R. (1828). Abrief account of Microsopical observations made in the months of June, July &Aug1827, on particles contained in the pollen of plants and the general existence of active molecules in organic & inorganic bodies,. *Phil. Mag, 4*(21), 161-173.

Brown, T. (1994). *Historical first patents: the first United States patent for many everyday things.* Michigan: University of Michigan: Scarecrow Press.

Bruce, ,. H. (1936). *1936- Nostalgic Australian photographs*. Retrieved August 2013, from Nostalgic Australian photographs: http://nla.gov.au/nla.pic-vn5125990

Cabinet Office, UK. (2013, February 20). *Resilient communications*. Retrieved March 31, 2013, from Gov.UK: https://www.gov.uk/resilient-communications

Cai, J., Shen, X., & Mark, J. W. (2003). Downlink Resource Management for Packet Transmission in OFDM Wireless Communication Systems. *IEEE Global Telecommunications Conference*, *6*, pp. pp2999-3003.

Camp, T., Boleng, J., & Davies, V. (2002). Survey of Mobility Models for Ad hoc Network Research . *Wireless Communication & Mobile Computing (WCMC) Special Issue Mobile Ad Hoc Networking Research, Trends and Application Vol 2*, (pp. 483-502).

Campell, ,. T. (1998). *QoS-aware Middleware for Mobile Multimedia Communications.* Retrieved December 28, 2004, from Columbia University: http://comet.ctr.columbia.edu/~campbell/papers/multimedia98.pdf

Cao, Y., & Li, V. (2001, Jan). Scheduling algorithms in broadband wireless networks. *Proceedings of the IEEE, General Topics for Engineers (Math, Science & Engineering) | Engineering Profession, Vol: 89*(1), pp76 - 87. Retrieved from http://ieeexplore.ieee.org/servlet/opac?punumber=5

Cellan-Jones, R. (2008, March 17th). Talking To Sir Tim. UK.

Chen, J., Jiao, W., & Wang, H. (2005). A service flow management strategy for IEEE 802.16 broadband wireless access systems in TDD mode. *IEEE International Conference on Communications. 5*, pp. pp 3422 - 3426. IEEE.

Chen, Y., Deng, D., Hsu, Y., & Wang, S. (2009). Enhanched Uplink Scheduling Algorithm for Video Traffic Transmission in IEEE 802.16 BWA Systems. *International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly,*, (pp. pp. 1330-1334). Leipzig.

Chu, G., Wang, D., & Mei, S. (2002). A QoS architecture for the MAC protocol of IEEE 802.16 BWA system. *IEEE Conference on Communications, Circuits, and Systems West Sino Expositions* (pp. PP 435 - 439). IEEE.

Cisco networking. (2003). Internetworking Technologies handbook, Cisco Press; 4 ed,. Cisco.

Dai, L., & Zhao, D. (2007). Uplink Scheduling for Supporting Real Time Voice Traffic. *The Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness & Workshops.* Vancouver.

Daniel, L., Florian, S., & Wolfgang, W. (2011). *Random Walks, Boundaries and Spectra* (Vol. Vol. 64). Springer.

Davis, J., Eisenhardt, K., & Bingham, C. (2007). Developing Theory Through Simulation Methods. *Academy of Management Review, 32*(2), 480-499.

Demers, A., Keshaw, S., & Shenker, S. (1989). Analysis and Simulation of a fair queuing algorithm. *SIGCOMM*, (pp. pp1-12).

Denning, P. J., Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J., & Young, P. R. (1989). COMPUTING AS A DISCIPLINE. *Communications of the ACM, 32*(1), 9-23.

Dhrona, P., Abu Ali, N., & Hassanein, H. (2008). A Performance Study of Scheduling Algorithms in Point-to-Multipoint WiMAX networks. *IEEE 33rd conference on local computer networks* , pp843 - 850.

Dodig-Crnkovic, G. (nd). Scientific Methods in Computer Science. Västerås, Sweden.

Dunbar, R. (1996). *Grooming, Gossip and the Evolution of Language.* London: Faber and Faber.

Einstein, A. (1956). *Investigation on the Theory of Brownian Movement.* New York: Dover.

environmentalgraffiti. (2012). *Paleolithic cave painting: Lascaux – Hall of Bulls / panorama*. Retrieved Auguust 12, 2012, from environmentalgraffiti: http://www.environmentalgraffiti.com/sciencetech/what-the-lascaux-cave-paintings-tell-us-about-how-our-ancestors-understood-the-stars/15506

Facebook. (2004). *facebook*. Retrieved August 28, 2012, from facebook: www.facebook.com

Fang, I. (1997). *A History of Mass Communication, Six Information Revolutions.* Boston, USA: Focal Press.

Farrell, J., & Klemperer, P. (2007). Coordination and Lock-In: Competition with Switching Costs and Network Effects. In M. Handbook of Industrial Organization, & R. Porter, *Handbook of Industrial Organization* (Vol. Volume 3, pp. pp 1967-2072). Elsevier.

Fattah, H., & Leung, C. (2002, Oct.). An overview of scheduling algorithms in wireless multimedia networks. *IEEE Wireless Communications, 9*, pp. 76–83.

Federation of Small Businesses. (2014). *The fourth utility: Delivering universal broadband connectivity for small businesses across the UK.* UK: www.fsb.org.uk.

Feyeraband, P. (2000). *Against Method.* London: Verso.

Floyd, S., & Allman, M. (2008, July). *RFC 5290 Simple Best-Effort Traffic.* Retrieved June 30, 2011, from IETF Network Working Group: http://tools.ietf.org/html/rfc5290

Franciszek, G., Paszkiewicz, A., & Bolanowski, M. (2013). Computer Networks as Complex Systems in Nonextensive Approach. *COMPUTER SCIENCE, 21(2)*, 31-44.

Freeman, R. L. (2005). *Fundamentals of Telecommunications 2nd Ed.* New Jersey: Wiley & Sons.

Ge, Y., & Kuo, G.-S. (2006). An Efficent Admission Control Scheme for adaptive Multimedia services in IEEE 802.16e Networks. *IEEE 64th Vehicular Technology Conference*, (pp. pp 1-5).

Geograph. (2005). *TG1128 : Milestone in the hedge near Corpusty, on Holt Road*. Retrieved August 02, 2012, from Geograph: http://www.geograph.org.uk/photo/528426

Ghosh, A., Wolter, D., Andrew, J., & Chen, R. (2005). Broadband wireless access with WiMax/802.16: current performance benchmarks and future potential. *Communications Magazine, IEEE, Vol. 43, No. 2*, pp. 129-136.

Gilbert, ,. E. (1960). Capacity of a burst-noise channel. *Bell Systems Technical Journal, 39*, pp 1253–1266.

Glomosim. (2011, June). Glomosim. http://pcl.cs.ucla.edu/projects/glomosim/.

Goldsmith, A. (2005). *Wireless Communications.* Cambridge University Press.

Goldsmith, A. J., & Varaiya, P. P. (1997). Capacity of fading channels with channel side information,”. *IEEE Transactions on Information Theory, IT-43*, pp 1896–1992.

Grigorik, I. (2012, July 26). *Latency: The New Web Performance Bottleneck*. Retrieved August 28, 2013, from igvita.com: http://www.igvita.com/2012/07/19/latency-the-new-web-performance-bottleneck

Grimes, T. R. (1990). Truth, Content, and the Hypothetico-Deductive Method. *Philosophy of Science , Vol. 57, No. 3*, pp. 514-522.

Gunasekaran, V., & Marmantzis, F. C. (2005). Affordable Infrastructure for Deploying WiMAX. *IEEE 62nd Semiannual Vehicular Technology Conference. 5*, pp. pp 2979 - 2983 . IEEE.

Hailey, D. (2009). *Theories of Communication, Language, and Human Discourse.* Retrieved August 28, 2012, from http://imrl.usu.edu/6890/TheoryofCommunication.pdf

Hamdi, N. (2007, Nov 15). Adaptive Max SNR Packet Scheduling for OFDM Wireless Systems. *Wireless Personal Communications, 46*(2), pp 223-232.

Haseeb, A., & Tralli, V. (2011). QoS Performance Analysis for VoIP Traffic in Heterogeneous Networks with WiMAX Access. *13th International Conference on Advanced Communication Technology (ICACT)*, (pp. pp 960 - 965). Seoul.

Hassel, V. (2007). *Design Issues and Performance Analysis for Opportunitistic Scheduling Algorithms in Wireless Networks.* Retrieved Dec 2010, from http://urn.kb.se/resolve?urn=urn:nbn:no:ntnu:diva-1202

Hassel, V., Alouini, M. S., Gesbert, D., & Øien, G. E. (2005). Exploiting multiuser diversity using multiple feedback thresholds.". *IEEE Vehicular Technology Conference.* Sweden: IEEE.

Helbing, D. (2001, Oct). Traffic and related self-driven many-particle systems. *Review of Modern Physics*, 1067-1141.

Held, G. (1999). *Understanding Data Communications 6th Edition.* New Riders Publishing. Retrieved from www.technet.microsoft.com: http://technet.microsoft.com/en-us/library/bb726929.aspx

Henry, S., & Woods, J. (2002). *Probability and Random Processes with Applications to Signal Processing* (3rd ed.). New Jersey: Prentice Hall.

History by Zim. (2012, April). *Bell System Switchboard, 1943.* Retrieved May 28, 2013, from History by Zim, Beyond the textbooks: http://www.historybyzim.com/2012/04/bell-system-switchboard-1943/

Holtzman, ,. J. (2001). Asymptotic analysis of proportional fair algorithm. *IEEE International Symposium on Personal, Indoor and Mobile Radio, 2*, pp33-37.

Hong, X., Gerla, M., Pei, G., & Chiang, C. (1999). A Group Mobility Model for Ad Hoc Wireless Network. *ACM International Workshop on Modelling and Simulation of Wireless and Mobile Systems (MSWiM'00).* ACM.

Hong, X., Gerla, M., Peri, G., & Chiang, C. (1999). A group mobility model for ad hoc wireless networks. *ACM International Workshop on Modelling and simulation of Wireless and Mobile Systems (MSWiM).* ACM.

Huston, G. (2000, November). *RFC 2990 - Next Steps for the IP QoS Architecture*. Retrieved June 28, 2010, from The Internet Engineering Task Force (IEFT): http://www.ietf.org/rfc/rfc2990.txt

hyperphysics. (2011, May). *Fast Fourier Transforms.* Retrieved February 21st, 2011, from hyperphysics: http://hyperphysics.phy-astr.gsu.edu/hbase/math/fft.html

IEEE . (2004). *Air interface for fixed broadband wireless access systems, IEEE 802.16 Standard.* IEEE.

IEEE . (2012). *IEEE Standards Association.* Retrieved April 26, 2014, from IEEE 802.16™: BROADBAND WIRELESS METROPOLITAN AREA NETWORKS (MANs): http://standards.ieee.org/about/get/802/802.16.html

IEEE WG802.11 - Wireless LAN Working Group. (2005). IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Spec. Retrieved from http://standards.ieee.org/findstds/standard/802.11e-2005.html- 25.6KB - IEEE SA

IEEE. (1997). 802.11-1997 IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE.

IEEE. (2003). *Part 11: Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) specifications.* IEEE.

IEEE. (2009, Sept 7). *802.11n-2009 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughp.* Retrieved April 26, 2014, from IEEE Standards Association: http://standards.ieee.org/findstds/standard/802.11n-2009.html

IEEE. (2011). *IEEE.* Retrieved January 2nd, 2011, from IEEE: https://sbwsweb.ieee.org/ecustomercme_enu/start.swe?SWECmd=GotoView&src=0&Join=n&SWEView=Catalog+View+%28eSales%29_Main_JournalMags_IEEE&mem_type=Customer&HideNew=N&SWEHo=sbwsweb.ieee.org&SWETS=1303738205

IEEE. (2012). *Cooke and Wheatstone's Electric Telegraph*. Retrieved Jan 2013, from IEEE Global History Network: http://www.ieeeghn.org/wiki/index.php/Cooke_and_Wheatstone's_Electric_Telegraph

IEEE 802. (2014, Jan 14). *NEW IEEE 802.11ac™ SPECIFICATION DRIVEN BY EVOLVING MARKET NEED FOR HIGHER, MULTI-USER THROUGHPUT IN WIRELESS LANS.* Retrieved April 26, 2014, from IEEE-STANDARDS ASSOCIATION: http://standards.ieee.org/news/2014/ieee_802_11ac_ballot.html

IEEE 802.16e-2005, IEEE Std 802.16-2004/Cor1-2005. (2006). *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2:Physical and Medium Access Control Layers for Combined Fixed a Operation in Licensed Bands and Corrigendum 1.* IEEE. Retrieved from http://ieeexplore.ieee.org/servlet/opac?punumber=10676

IEEE. (nd). *IEEE 802.11a - Working Group*. Retrieved November 28, 2013, from IEEE 802.11: grouper.ieee.org/groups/802/11/

IEEE. (nd). *IEEE.ca.* Retrieved March 28, 2014, from An Unsung hero: Reginald Fessenden, the Canadian inventor of radio telephony: http://www.ieee.ca/millennium/radio/radio_unsung.html

IEEE Standard 802.16.2. (2004). *IEEE Standard 802.16.2-2004 IEEE Recommended Practice for Local and Metropolitan Area Networks.* IEEE.

IEEE Std 802.11e-2005. (2005, November). *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements.* IEEE.

IEFT. (1998, August). *Interoperation of Controlled-Load Service and Guaranteed Service with ATM.* Retrieved from IEFT 2381 Network Working Group: https://tools.ietf.org/html/rfc2381

Ingen-Housz, J. (1784). *Vermischte Schriften Physisch Medicinischen Inhalts* (Vol. 2). Vienna: Wapler.

Intel Information Technology-Gerard Smyth. (2006). *Wireless Technologies and e-Learning: Bridging the.* United States: Intel Information Technology.

International Softswitch Consortium. (2008). Retrieved December 7th, 2008, from International Softswitch Consortium: http://www.softswitch.org/

International Telecommunication Union - Telecommunication. (1996, feb 6). *International Telecommunication Union - Telecommunication G.114*. Retrieved June 1st, 2010, from International Telecommunication Union - Telecommunication: http://www.itu.int/itudoc/itu-t/aap/sg12aap/history/g.114/

International Telecommunication Union. (2000, September 27). A Canadian welcome to WTSA-2000. *ITU News Issue No. 10/99*, p. 8. Retrieved from International Telecommunication Union: http://www.itu.int/itunews/issue/2000/07/host.html

Internet World Stats. (2013, August 28). *World Internet Users and Population Stats*. Retrieved August 28, 2013, from Internet World Stats: http://www.internetworldstats.com/stats.htm

ITU. (1998, Nov). *Recommendation BT.1363.* Retrieved April 28, 2014, from International Telecommuncications Union: http://www.itu.int/rec/R-REC-BT.1363/en

ITU. (2004). *Handbook Quality of Service and Network Performance Edition 2004.* Geneva: International Telecommunications Union.

ITU. (2005). *Telecommunication indicators update.* Retrieved March 3rd, 2010, from International Telecommunication Union: www.itu.int/ITU-D/ict/statistics/

ITU. (2013, Dec 6). *ITU - Statistics - ICT Facts and Figures 2013*. Retrieved Dec 28, 2013, from ITU: http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx

ITU(a). (2005, Dec). *Cellular Standards for the Third Generation.* Retrieved from About mobile technology and IMT-2000: http://www.itu.int/osg/spu/imt-

2000/technology.html#Cellular%20Standards%20for%20the%20Third%20Generation

ITU-R. (2013, Jan 17). *ITU Global Standard for International Mobile Telecommunications 'IMT-Advanced'.* Retrieved from www.ITU.int/ITU-R/index.asp?category=information&link=imt-advanced&lang=en

ITU-T Recommendation E.800. (2008). *Terms and definitions related to quality of service and network performance including dependability.* Retrieved from http://www.itu.int/rec/T-REC-E.800

ITU-T Recommendation I.150. (1999, February). *B-ISDN asynchronous transfer mode functional characteristics.* Retrieved June 2011, from ITU-T: http://www.itu.int/

ITU-T Study Group 17. (2009, 10 29). *ITU-T X.902 (10/2009).* Retrieved April 28, 2012, from ITU-Telecoms: http://handle.itu.int/11.1002/1000/10242

ITU-T study group 2. (2007, September). *Teletraffic Engineering Handbook.* Retrieved 01 09, 01, from ITU-T: http://www.com.dtu.dk/teletraffic/handbook/telenook.pdf

Jain, K., & Tewari, R. (2012). Influence of Mobility Models in Performance Evaluation of MANET Routing Protocols. *3rd International IT Summit Confluence 2012 - The Next Generation Information Technology Summit.* Nodia, India: FCS.

Jain, R., Lelescu, D., & Balakrishnan, M. (2005). Model T: and Empirical Model for User Registration pattrens in a Campus Wireless LAN. *MobiCom'05* (pp. 170-184). ACM.

Jain, R., So-in, C., & Tamimi, A. (2009). System Level Modeling of IEEE 802.16e Mobile WiMAX Networks: Key Issues. *IEEE Journal on Selected Areas in Communications* , pp. 156-171.

Jalali, A., Padovani, R., & Pankaj, R. (2002). Data Throughput of CDMA-HDR, a High Efficiency High Data Rate Personal Communication Wireless System. *IEEE 51st Vehicular Technology Conference* (pp. pp 1854-1858). Tokyo, Japan: IEEE.

Jiang, C.-H., & Tsai, T.-C. (2006). Token Bucket Based CAC and Packet Scheduling for IEEE 802.16 Broadband Wireless Access Networks. *IEEE 3rd Consumer Communications and Networking Conference*, (pp. pp183-187).

Johansson, M. (2004). Diversity-enhanced equal access - considerable through throughput. *IEEE Workshop on Signal Processing Advances in Wireless Communications.* Lisbon, Portugal.

Johnson, D., & Maltz, D. (1996). Dynamic source routing in ad hoc wireless networks. In T. Lmielinki, & H. Korth, *Mobile Computing* (pp. 153-181). Klumer Academic Publishers.

Kalikivaya, S., Misra, I. S., & Saha, K. (2008). Bandwidth and Delay Guaranteed Call Admission Control scheme for QoS Provisioning in IEEE 802.16e Mobile WiMAx. *IEEE Global Telecommunications Conference* (pp. pp 1 - 6). IEEE.

Kappler, C., Fu, X., & Schloer, B. (2011). *RFC2381 Interoperation of controlled-loadservice.* IETF.

Karp, B., & Kung, T. (2000). GPSR: Greedy perimeter stateless routing. *International Conference on Mobile Computing and Networking (Mobicom)* (pp. 43–254.). ACM/IEEE.

Katevenis, M., Sidiropoulos, S., & Courcoubetis. (1991). Weighted round-robin cell multiplexing in a general-purpose ATM switch chip. *IEEE Journal on Selected Areas in Communications*, pp 1265–1279.

Kim, M., & Kotz, D. (2005). Modeling Users' Mobility among WiFi Access Points. *WiTMeMo'05.*

Kim, M., Kotz, D., & Kim, S. (2006). Extracting a Mobility Model from Real User Traces. *25th IEEE International Conference on Computer Communications* (pp. pp.1,13). Barcelona, Spain: IEEE.

Knopp, R., & Humblet, P. A. (1995). Information capacity and power control in single cell multiuser communications. *IEEE Int. Conference on Communications*, (pp. pp 331–335,). USA.

Kollewin Blog. (2009, Sept 23). *Round Robin*. Retrieved April 2, 2010, from Kollewin Blog: http://www.kollewin.com/blog/round-robin-scheduling/

Kuenyoung, K., Hoon, K., & Youngnam, H. (2002). A proportionally fair scheduling algorithm with QoS and priority in 1xEV-DO. *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications;. vol.5*, pp. pp.2239,2243. IEEE.

Kulkarni, S. S., & Rosenberg, C. (2003). Opportunistic scheduling policies for wireless systems with short term fairness constraints. *IEEE Global Communications* (pp. pp 533–537). San Francisco, CA, USA: IEEE.

Lakkakorpi, J., Sayenko, A., & Moilanen, J. (2008). Comparison of Diffrent Scheduling Algorithms for WiMAX Base Station: Deficit Round-Robin vs. Proportional Fair vs. Weighted Deficit Round-Robin. *Wireless Communications and Networking Conference* (pp. pp1991-1996). IEEE.

Law, A. (2007). *Simulation Modelling & analysis.* singapore: Mcgraw-Hill.

Law, A. (2007). *Simulation Modelling & analysis.* Singapore: Mcgraw-Hill.

Law, A. M. (2003). How to Conduct a Successful Simulation Study. *2003Winter Simulation Conference*, (pp. 66-70). New Orleans.

Lee, J. K., & Hou, J. (2006). Modelling Steady-State and Transient Behaviours of Mobility: Formulation, Analysis, and Application. *MobiHoc'06.* ACM.

Lee, S., Gerla, M., & Chiang, M. (1999). On-demand Multicast Routing Protocols. *IEEE Wireless Communications and Networking (WCNC'99).* IEEE.

Liang, B., & Haas, Z. (1999). Predictive distance-based mobility management for PCS networks. *IEEE Computer and Communication Societies (INFOCOM).*

Lin, P., Ngo, H., & Qiao, C. (2006). Minimum Cost Wireless Broadband Overlay Network. *International Symposium on a World of Wireless Mobile and Multimedia networks*, (pp. pp. 1–7).

Lin, Y.-N., Lin, Y.-D., & Lai, Y.-C. (2009). Highest Urgency First (HUF): A latency and modulation aware bandwidth allocation algorithm for WiMAX base stations. *Computer Communications , 32*, pp 332–342.

Liu, N., Li, X., & Pei, B. (2005). Delay character of a novel architecture for IEEE 802.16 systems. *6th Conference on Parallel and Distributed Computing, Applications and Technologies*, (pp. pp.293-296).

Lu, J., Ma, M., & Gong, Y. (2009). A Cross-Layer MAC Protocol and an Opportunistic Scheduling with APC for QoS Support in WiMAX.

Ludkovski, M. (2007). *Browian morion Notes*. Retrieved 2012, from Statistics & Applied Probability, University of California: www.pstat.ucsb.edu/ludkovski/bmnotes.pdf

Mähönen, P., Saarinen, T., & Shelby, Z. (2001, May). Wireless Internet over LMDS: Architecture and experimental implementation. *IEEE Communications Magarine, vol 5 no 39*, pp. pp.126-132.

Mahoney, M. S. (1992). COMPUTER SCIENCE: The Search for a Mathematical Theory. In J. Echeverria, A. Ibarra, & T. Mormann, *The Space of Mathematics* (pp. 347-361). Berlin/New York: De Gruyter.

Marconi, G. (1909, December 11). *Nobel Prize.* Retrieved March 28, 2014, from Wireless telegraphic communication, Nobel Lecture: http://www.nobelprize.org/nobel_prizes/physics/laureates/1909/marconi-lecture.pdf

Marks, R. (2006). IEEE 802.16 WirelessMAN Standard: Myths and Facts. *Wireless Communications Conference.* Washington, DC: ieee802.org.

Massachusetts Institute of Technology . (2004, August). *Inventor of the Week Archive - Johann Gutenberg*. Retrieved Dec 27, 2013, from Lemonson - MIT: http://web.mit.edu/invent/iow/gutenberg.html

Meucci, F., Pierucci, L., & Cerutti, I. (2010). Performance of Dynamic Service Addition in. *5th International Sypnposium on Wireless Pervasive Computing* (pp. pp612-617). Modens, Italy: IEEE.

Microsoft. (2011, March 31). *Networking and Access Technologies.* Retrieved June 30, 2011, from www.microsoft.com: http://technet.microsoft.com/en-us/network/bb530836

Miller, S. G. (2012). *Arete: Greek Sports from Ancient Sources.* California USA: University of California Press.

Montagu, A. (1979). *The Human Connection.* Msgraw-Hill.

Mörters, P., & Peres, Y. (2008). *Brownian Motion.* Retrieved May 20, 2011, from Microsoft: http://research.microsoft.com/en-us/um/pepole/peres/brbook.pdf

Nagle, J. (1987). On packet Switches with infinite storage. *Transaction Communications. 35.* IEEE.

NASSA. (nd). *Scientists and Electromagnetic Waves: Maxwell and Hertz*. Retrieved November 28, 2013, from The electronmagnetic spectrum: http://science.hq.nasa.gov/kids/imagers/ems/consider.html

Neelakanta, P. S. (2000). *A Textbook on ATM Telecommunications, Principles and implementation.* CRC Press.

Newell, A. (1985). Artificial Intelligence - Where Are We? In D. G. Bobrow, & P. J. Hayes, *Artifical Intelligence* (Vol. 25).

Newell, Perlis, & Simon. (1967). Computer Science. *Science, Vol. 157*(3795), pp. 1373-1374.

Nickinson, P. (2010, January 18). *HTC A9292 may be Sprint WiMAX 'Supersonic'*. Retrieved June 30, 2011, from androidcentral: HTC A9292 may be Sprint WiMAX 'Supersonic' | Android Central

Nie, W., Xiong, N., & Wang, H. (2010). A Novel Hybrid uplink Bandwidth Scheduler in WiMAX Real Time Communication Networks. In IEEE (Ed.), *5th International Conference on Ubiquitous Information Technologies and Applications (CUTE),* (pp. pp 1 - 6). Sanya : IEEE.

NS3. (2011, July). NS3. Retrieved from http://www.nsnam.org/documentation/

OfCom. (2013, March 1). *880Mhz & 2.6Ghz Combined Award.* Retrieved from stakeholders.ofcom.org: stakeholders.ofcom.org.uk/spectrum-awards-archive/completed-award/800Mhz-2.6ghz/

Ohrtman, F. (2005). *WiMAX Handbook :building 802.16 Wireless Networks.* Berkely, USA: McGraw-Hill.

Parekh, A. K., & Gallager, R. G. (1993). A generalized Processor Sharing Approach to Flow Control in Intregreted Servicesnetworks: The Single node Case. *IEEE/ACM Transactions on Networking*, *1 No 2*, pp. pp334-357.

Parry, R. (2011). *The Ascent of Media: From Gilgamesh to Google Via Gutenberg.* London: Nicholas Brealey Publishing.

Peart, A., & Adda, M. (2011). A QoS Real Time Bandwidth Redistribution Transmission Algorithm in WiMAX. *Computer Science & Information Systems* (p. 21). Athens, Greece: ATINER.

Peart, A., & Adda, M. (2013). Quality of service in WiMAX: real world aspects of social & environmental influences on mobility. *9th Annual International Conference on Information Technology & Computer Science* (p. pp 428). Athens Greece: ATINER.

Peart, A., Adda, M., & Goodman, A. (2012). A QoS Real Time Bandwidth Redistribution Transmission Algorithm in WiMAX. In Y. Papadopoulos., & P. P, *Enterprise Management Information Systems* (pp. pp 63-72). Athens: ATINER.

Peart, A., Lockett, A., & Adda, A. (2013). Predicting mobile network bandwidth fluctuation to enhance video stream service qualityImplementation of location-based, dynamic transmission rate-limit control. *Science and Information Conference 2013* (pp. pp 829-835). London, UK: IEEE Xplore.

Pelcat, M. A. (2013). *Physical Layer Multi-Core Prototyping.* New York: Springer.

Philpot, D., Beaton, B., & Whiteduck, T. (2014). First Mile Challenges to Last Mile Rhetoric: Exploring the Discourse between Remote and Rural First Nations and the Telecom Industry. *The Journal of Community Informatics, 10(2).*, 10-16.

Plummer, L. (2012, Jan 21). *What is 4G*. Retrieved from Pocket lint: www.pocket-lint.com/news/108087-what-is-4g-lte-wimax

Quantumfreak. (2013). *Motion of Molecules*. Retrieved January 2013, from Quantumfreak: http://quantumfreak.com/motion-of-molecules/

Radio-Electronics. (2011). *What is QAM - Quadrature Amplitude Modulation*. Retrieved July 1, 2011, from www.radio-electronics.com: http://www.radio-electronics.com/info/rf-technology-design/pm-phase-modulation/what-is-qam-quadrature-amplitude-modulation-tutorial.php

Ramachandran, S. (2004). *Link Adaptation Algorithm and Metric for IEEE Standard 802.16.* USA - Virginia: Virginia Polytechnic Institute and State University.

Ramachandran, S., Bostian, C. W., & Midkiff, S. F. (2005). A Link Adaptation Algorithm for IEEE 802.16. *IEEE Wireless Communications and Networking Conference* (pp. pp 1466-1471). /new Orleans, USA: IEEE.

Rentel, C. H., Krzymien, W. A., Darian, B., Vanghi, V., & Elliott, R. (2002). Comparative forward link traffic channel performance evaluation of HDR and 1XTREME systems. *IEEE Vehicular Technology Conference,* (pp. pp 160-164). IEEE.

Resta, G., & Santi, P. (2006). The QoS-RWP Mobility and User Behaviour Model for Public Area Wireless Networks. *MSWIM'06.* Torremolinos, Spain: ACM.

Revuz, D., & Marc, Y. (2008). *Continuous Martingales and Brownian Motion* (3rd ed.). Springer.

Rheingold, H. (2000). *Tools for thought: The History and Future of Mind-Expanding Technology.* MIT Press.

Richard, P. (Fall 2008 Edition). Episteme and Techne. In E. N. (ed.), *The Stanford Encyclopedia of Philosophy.* Retrieved 2010, from http://plato.stanford.edu/archives/fall2008/entries/episteme-techne/

Ringland, J. (2010). *System Science of Virtual Reality: Toward the Unification Empirical and Subjective Science .* lulu.

Rodrigues. (2006, November 11). *WiFi and Security Blog*. Retrieved June 30, 2011, from reseau-wifi.blogspot: www.reseau-wifi.blogspot.com

Rohde & Schwarz. (2014). *Mobile WiMAX™ Throughput Measurements Application Note .* Retrieved from Rohde & Schwarz: http://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1sc10/1SP10_2e.pdf

Rosenfeld, J. (2000, March 1st). Here's an Idea! Ashburn, Virginia.

Rubin, I., & Choi, C. (1997). Impact of the Location Area Structure on The Performance of Signalling Channels in Wireless Cellular Networks. *IEEE Communications Magazine* (pp. 108-115). IEEE.

Ruskin, J. (1819 - 1900). *BrainyQuote.com.* Retrieved June 30, 2011, from http://www.brainyquote.com/quotes/authors/j/john_ruskin_4.html

Salus, P. H., & Vinton, G. (1995). *Casting the Net: From ARPANET to Internet and Beyond...* Boston, MA, USA: Addison-Wesley Longman Publishing Co.

Samsung. (2009). *Product Review.* Retrieved June 30, 2011, from Samsung: www.samsung.com/uk/

Sánchez, M., & Manzoni, M. (2001). ANEJOS: a Java based simulator for ad hoc networks. *Future Generation Computer System*, (pp. 573-583).

Sanchez, M., & Manzoni, P. (Jan 1999). A Java-Based Ad-Hoc Networks simulator. *SCS Western Multi conference, Web based simulation track.* San Francisco.

Savage, J. E. (1998). *Exploring the power of Computing.* Addison-Wilseley.

Sayenko, A., Alanen, O., & Hämäläinena, T. (2008). Scheduling solution for the IEEE 802.16 base station. *The International Journal of Computer and Telecommunications Networking. 52*, pp. pp 96-115. Elsevier.

Schummer, J. (2001). Aristotle on Technology and Nature. *Philosophia Naturalis, 38*, 105-120.

Shepard, A. (2004, April 28). Hybrid Change Makes WLAN QoS Come to Life. *CommsDesign*.

Shetiya, H., & Sharma, V. (2005). 1st ACM workshop on Wireless multimedia networking and performance modeling . *Wireless Multimedia Networking and Performance Modeling* (pp. pp 140 - 149). Montreal: ACM.

Shreedhar, M., & Varghese, G. (1996). Efficient fair queueing using Deficit Round-Robin. *IEEE/ACM Transactions on Networking. 4*, pp. pp 375-385. IEEE/ACM.

Soldatos, J., Vayias, E., & Kormentzas, G. (2005). On the building blocks of quality of service in heterogeneous IP networks. *IEEE Communications Surveys & Tutorials*, pp 69 - 88 .

Soon Young Yoon. (2004, September 10). Telecom R&D Center. Retrieved Decemeber 12th, 2010, from SAMSUNG Electronics Co., Ltd.: http://www.itu.int/ITU-D/imt-2000/Meetings/Busan/Session3_Yoon.pdf

Stiladis, D., & Varma, A. (1998). Latency rate servers: A general mode for analysis of traffic scheduling algorithms. *IEEE/ACM Transactions on Networking, 6*(5), pp.611,624.

Stolyar, A. L., & Ramanan, K. (2001). *Largest Weighted Delay First Scheduling: Large Deviations and Optimality.* Annals of Appl. Prob.

Sweeney, D. (2006). WiMax Operator's Manual: Building 802.16 Wireless Networks. Apress.

Tanenbaum, A. S., & Wetherall, D. J. (2010). *Computer Networks 5th edition.* USA.

The British Musemum. (2014, April 28). The Rosetta Stone - Rosetta, Egypt. 203BC. London, UK.

The Guardian; Moss, S. (2013, July 10). Final telegram to be sent. STOP. London, UK.

The Nobel Prize. (2013). *The Nobel Prize in Physics 1909 - Guglielmo Marconi - Biographical*. Retrieved Dec 27, 2013, from The Nobel Prize.Org Nobel Media AB 2013. Web.: http://www.nobelprize.org/nobel_prizes/physics/laureates/1909/marconi-bio.html

The Science Museum. (2004). *The Growth of the Railways*. Retrieved December 28, 2013, from Making the Modern World: http://www.makingthemodernworld.org.uk/learning_modules/history/04.TU.03/?section=11

Tichy, W. F. (1998, May). Should Computer Scientists Experiment More? *Computer, Vol 31*(5).

Tsang, K. F., Lee, L. T., Tung, H. Y., Lam, R., Sun, Y. T., & Ko, K. T. (2007). Admission Control Scheme for Mobile WiMAx Networks. *IEEE International Symposium on Consumer Electronics, 2007.*, (pp. pp 1 - 5).

Tung, H. Y., Tsang, ,. K., Lee, L. T., & Ko, K. T. (2008). QoS for Mobile WiMAX Networks: Call Admission Control and Bandwidth Allocation. *5th IEEE Consumer Communications and Networking Conference (CCNC 2008)*, (pp. pp 576 - 580).

Twitter. (2006). *twitter*. Retrieved August 28, 2012, from twitter: https://twitter.com

Twitter. (2014, May 31). *Counting Characters*. Retrieved from Twitter.com: https://dev.twitter.com/docs/counting-characters

University of North Texas Libraries, The Portal to Texas History. (nd). *Telephone Switchboard, Richardson, Texas, Photograph, ca. 1900.* Retrieved July 17, 2014, from The Portal to Texas History: http://texashistory.unt.edu/ark:/67531/metapth3137

Vilches, J. (2010, April 29). *Tech Spot.* Retrieved March 28, 2014, from Everything you need to know about 4G Technology: http://www.techspot.com/guides/272-everything-about-4g/

Vinay, K., Sreenivasulu, ,. N., Jayaram, D., & Das, D. (2006). Performance Evaluation of End-to-end Delay by Hybrid Scheduling Algorithm for QoS in IEEE 802.16 Network. *International Conference on Wireless and Optical Communications Networks* (pp. pp 1-5). Bangalore: IEEE.

VoIPforo.com. (nd). *VoIP protocols.* Retrieved 04 02, 08, from http://www.en.voipforo.com/QoS/QoS_jitter.php

Wang, H., Li, W., & Agrrawal, D. P. (2005). Dy-nanic admission control and QoS for IEEE 802.16 Wireless MAN. *Wireless Telecommunications Symposium*, (pp. pp 60-66).

Wang, L., Liu, F., ji, Y., & Ruang-chaijatupon, N. (2007). Addission Control for Non-preprovisioned Service Flow in Wireless Metropolitan Area Networks. *4th European Conference Universal Multiservice Networks*, (pp. pp 243-249).

Wang, Z. (2001). *Internet QoS: Architecture and Mechanisms for Quality of Service.* Morgan.

Wegner, P. (1976). Research Paradigms in computer science. *2nd International Conference onSoftware Engineering.* San Francisco.

Weisstein, E. W. (1998). The CRC Concise Encylopedia of Mathematics. CRC Press.

Whelan, M. (2010). *Edison Tech Cente*. Retrieved March 28, 2014, from Engineering Hall of Fame - Joseph Henry: http://www.edisontechcenter.org/JosephHenry.html

Wiethőlter, S., & Hoene, C. (2003). Design and Verification of an IEEE 802.11e ECDF Simulation Model in NS 2.26. *telecoms network group*, pp. 3-19.

WiMAX Forum. (2001-2014). *WiMAX Forum*. Retrieved July 17, 2014, from Technical Specifications: http://www.wimaxforum.org/

Wimax Forum. (2011). *About the WiMAX Forum*. Retrieved June 30, 2011, from wimaxforum.org: http://www.wimaxforum.org/

Wongthavarawat, K., & Ganz, A. (2003). IEEE 802.16 BASED LAST MILE BROADBAND WIRELESS MILITARY NETWORKS WITH QUALITY OF SERVICE SUPPORT. *Military Communications conference. 2*, pp. pp 779 - 784. USA: IEEE.

Wongthavarawat, K., & Ganz, A. (2003). Packet Scheduling for QoS support in IEEE 802.16 Broadband Wireless Access Systems. *International Journal of Communication Systems, 16*, 81-96. doi:10.1002/dac.581

Wongthavarawat, K., & Ganz, A. (2003). Packet Scheduling for QoS Support in iEEE 802.16 broaddband wireless access systems. *International Journal of Communication Systems, Vol. 16, No. 1.*, pp. 81-96.

World Wide Web Consortium. (2013, August). *Tim Berners-Lee*. Retrieved August 28, 2013, from World Wide Web Consortium: http://www.w3.org/People/Berners-Lee/Overview.html

Worldstats. (2011). *Internat and Worldstas: usage and Popoulation Statistics.* Internet World Stats.

Yang, L., & Alouini, M. S. (2004). Performance analysis of multiuser selection diversity. *IEEE Int. Conference on Communications*, 3066–3070.

Yoon, J., Liu, M., & Noble, B. (2003). Random Waypoint Considered Harmful. *InfoCom.* San Francisco, CA: IEEE.

YouTube. (2015, April 20). *YouTube for Developers* . Retrieved from YouTube: https://www.youtube.com/yt/dev/index.html

YouTube. (2015, April 20). *YouTube Statistic*. Retrieved from YouTube: https://www.youtube.com/yt/press/statistics.html

Zhu, H. J., & Hafez, R. H. (2006). Novel Scheduling Algorithms for Multimedia. *IEEE Communications Society.* IEEE Communications Society.

Zonoozi, M., & Dassanayake, P. (1997). User Mobility Modelling and Characterization of Mobility Patterns. *IEEE Journal on Selected areas on Comunications, 15*(7), 1239-1252.

Appendices

FORM UPR16

Research Ethics Review Checklist

University of Portsmouth

1. **Please complete and return the form to Research Section, Quality Management Division, Academic Registry, University House, with your thesis, prior to examination**

2.

Postgraduate Research Student (PGRS) Informatio		**Stude**	**63950**
Candidate Name:	**Amanda Peart**		
Department:	**School of Con**	**First Supervis**	**Dr Mo Adda**
Start Date: (or progression date for Prof Doc students)	**Oct 2006**		

Study Mode and Route:	Part-time √ Full-time ❑	MPhil ❑ MD ❑ PhD √	Integrated Doctorate (NewRoute) ❑ Prof Doc (PD) ❑

Title of Thesis:	**Optimising Quality of Service Levels through Experimentation on Streaming Multimedia Applications using WiMAX.**
Thesis Word Count: (excluding ancillary data)	42 882

If you are unsure about any of the following, please contact the local representative on your Faculty Ethics Committee for advice. Please note that it is your responsibility to follow the University's Ethics Policy and any relevant University, academic or professional guidelines in the conduct of your study

Although the Ethics Committee may have given your study a favourable opinion, the final responsibility for the ethical conduct of this work lies with the researcher(s).

UKRIO Finished Research Checklist:
(If you would like to know more about the checklist, please see your Faculty or Departmental Ethics Committee rep or see the online version of the full checklist at: http://www.ukrio.org/what-we-do/code-of-practice-for-research/)

a) Have all of your research and findings been reported accurately, honestly and within a reasonable time frame?	**Y**
b) Have all contributions to knowledge been acknowledged?	**Y**
c) Have you complied with all agreements relating to intellectual property, publication and authorship?	**Y**
d) Has your research data been retained in a secure and accessible form and will it remain so for the required duration?	**Y**
e) Does your research comply with all legal, ethical, and contractual requirements?	**Y**

*Delete as appropriate

Candidate Statement:	
I have considered the ethical dimensions of the above named research project, and have successfully obtained the necessary ethical approval(s)	
Ethical review number(s) from Faculty Ethics Committee (or from NRES/SCREC):	
Signed:  ***(Student)***	**Date: 28/011/14**
If you have *not* submitted your work for ethical review, and/or you have answered 'No' to one or more of questions a) to e), please explain why this is so:	
The nature of this research does not include human participants in any form (inc personal data that is not in the public domain); nor does it encroach on sensitive topics. The research do involve any ethical issues relating to the integrity of the environment.	
Signed:  ***(Student)***	**Date: 28/011/14**

Random Walk Code snippet

```
#include "ns3/core-module.h"
#include "ns3/helper-module.h"
#include "ns3/mobility-module.h"
#include "ns3/simulator-module.h"

using namespace ns3;

static void
CourseChange (std::string foo, Ptr<const MobilityModel> mobility)
{
  Vector pos = mobility->GetPosition ();
  Vector vel = mobility->GetVelocity ();
  std::cout << Simulator::Now ()  << "," << mobility << ","  <<  pos.x << ","  <<
pos.y << std::endl;

}

int main (int argc, char *argv[])
{
  Config::SetDefault ("ns3::RandomWalk2dMobilityModel::Mode", StringValue
("Time"));
  Config::SetDefault ("ns3::RandomWalk2dMobilityModel::Time", StringValue
("10s"));
  Config::SetDefault ("ns3::RandomWalk2dMobilityModel::Speed", StringValue
("Constant:15.0")); // speed of the movement
  Config::SetDefault ("ns3::RandomWalk2dMobilityModel::Bounds", StringValue
("0|6000|0|6000"));
  //the boundary of the area that the nodes will move within

  CommandLine cmd;
  cmd.Parse (argc, argv);

  NodeContainer c;
  c.Create (1); //the number of nodes created to move around

  MobilityHelper mobility;
  mobility.SetPositionAllocator ("ns3::RandomDiscPositionAllocator",
                                 "X", StringValue ("0.0"),
                                 "Y", StringValue ("0.0"),
                                 "Rho", StringValue ("Uniform:1:30"));
  mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel",
                             "Mode", StringValue ("Time"),
                             "Time", StringValue ("10s"),
                             "Speed", StringValue ("Constant:15.0"),
                             "Bounds", StringValue ("0|6000|0|6000"));
```

```
//------------------------------------------------------------------------
  c.Create (1); //the number of nodes created to move around


  mobility.SetPositionAllocator ("ns3::RandomDiscPositionAllocator",
                                 "X", StringValue ("100.0"),
                                 "Y", StringValue ("100.0"),
                                 "Rho", StringValue ("Uniform:1:50"));
  //random variable which represents the radius of a position in a random disc.
Min=0.0|Max=200.0]

  /*mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel",
                             "Mode", StringValue ("Time"),
                             "Time", StringValue ("10s"),
                             "Speed", StringValue ("Constant:15.0"),
                             "Bounds", StringValue ("0|6000|0|6000"));*/
//------------------------------------------------------------------------
c.Create (1); //the number of nodes created to move around


  mobility.SetPositionAllocator ("ns3::RandomDiscPositionAllocator",
                                 "X", StringValue ("10.0"),
                                 "Y", StringValue ("10.0"),
                                 "Rho", StringValue ("Uniform:1:10"));
 /* mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel",
                             "Mode", StringValue ("Time"),
                             "Time", StringValue ("10s"),
                             "Speed", StringValue ("Constant:15.0"),
                             "Bounds", StringValue ("0|1000|0|1000"));*/
  mobility.InstallAll ();
  Config::Connect ("/NodeList/*/$ns3::MobilityModel/CourseChange",
                   MakeCallback (&CourseChange));

  Simulator::Stop (Seconds (100.0));

  Simulator::Run ();

  return 0;
}
```

Random Walk With IPV4

```
#include "ns3/core-module.h"
#include "ns3/common-module.h"
#include "ns3/node-module.h"
#include "ns3/helper-module.h"
#include "ns3/mobility-module.h"
#include "ns3/contrib-module.h"
#include "ns3/wimax-module.h"
#include <ostream>
#include <fstream>
#include <string>
#include <stdio.h>
#include "ns3/global-route-manager.h"
#include "ns3/gnuplot.h"


NS_LOG_COMPONENT_DEFINE ("wimaxIpV4Simulation");

using namespace ns3;

static double BytesRcv = 0;
//static double BytesSend = 0;
static uint32_t rec=0;
static uint32_t send=0;


static std::string imageName ;
static Gnuplot m_gnuplot;
static Gnuplot2dDataset m_output;///< needed to plot the total throughput

/////////////////////////////////////////////////////////////////////////////////

void ReceivedPacket (std::string context, Ptr<const Packet> p, const Address& addr)
{
  BytesRcv += p->GetSize ();
  rec++;

}

////////////////////////////////////////////////////////////////////
void SendPacket( std::string context, Ptr<const Packet> p)
{

std::cout << " Txx=" << Simulator::Now ().GetSeconds()<<std::endl;
 send++;
}
```

```
///////////////////////////////////////////////////////////////////////////

void Throughput (double interval)
 {
   double rate = (((  BytesRcv * 8.0) / 1000000)/interval);
    BytesRcv = 1;
    Time time = Simulator::Now ();
    m_output.Add (time.GetSeconds(),rate);
    Simulator::Schedule (Seconds (interval), &Throughput,interval);
std::cout << "The value of time is" << time << "node is " << interval << std::endl;
std::cout << " Txx=" << Simulator::Now ().GetSeconds()<<std::endl;
 }

///////////////////////////////////////////////////////////////////////////

std::string ImageName (int i)
 {
   std::string rootName ="net-measure_throughput";
    std::string run = "run-";
   std::string num[6]={"0","1","2","3","4","5"};
   std::string sufix =".png";
    std::string n = num[i];
    std::string imageName = rootName + sufix;
    return imageName;
 }

////////////////////////////////////////////////////////////////////////


///////////////////////////////////////////////////////////////////////////

int main (int argc, char *argv[])
{
  // default values
  int nbSS = 4, duration = 7, schedType = 1, delay = 1, npacket = 1;imageName =
ImageName (1);


  bool verbose = false;
  double interval = 1,  nspeed = 1 ;

  std::string context; Ptr<const Packet> p;
 ////////////////////////////////////////////////////////set up gnuplot variables for results

  m_gnuplot = Gnuplot (imageName);
```

```
  m_gnuplot.SetTitle ("Throughput vs Time");
  m_gnuplot.SetLegend ("Time(Second) ","Throughput [Mbps]");
  m_output.SetStyle (Gnuplot2dDataset::LINES_POINTS);

////////////////////////////////////////////////////////////////////////////////

  //instantiate scheduling type
  WimaxHelper::SchedulerType scheduler = WimaxHelper::SCHED_TYPE_SIMPLE;

  LogComponentEnable ("UdpClient", LOG_LEVEL_INFO);
  LogComponentEnable ("UdpServer", LOG_LEVEL_INFO);



  CommandLine cmd;
  cmd.AddValue ("nbSS", "number of subscriber station to create", nbSS);
  cmd.AddValue ("scheduler", "type of scheduler to use with the network devices",
schedType);
  cmd.AddValue ("duration", "duration of the simulation in seconds", duration);
  cmd.AddValue ("verbose", "turn on all WimaxNetDevice log components", verbose);
  cmd.AddValue ("delay", "duration of delay", delay);
  cmd.AddValue ("npacket", "random packets", npacket);
  cmd.AddValue ("speed", "speed of ss movement", nspeed);
  //cmd.AddValue ("bandwidth", "bandwidth per channel", bandwidth);
  cmd.Parse (argc, argv);

  //set schedule type is determined by the priority requirements

  switch (schedType)
    {
    case 0:
      scheduler = WimaxHelper::SCHED_TYPE_SIMPLE; //a simple priority based FCFS
scheduler this is ok for nrtps and BE
      break;
    case 1:
      scheduler = WimaxHelper::SCHED_TYPE_MBQOS; //a migration-based uplink
scheduler
      break;
    case 2:
      scheduler = WimaxHelper::SCHED_TYPE_RTPS; //real-time polling service (rtPS)
scheduler this is ok for ertps and rtps
      break;
    default:
      scheduler = WimaxHelper::SCHED_TYPE_SIMPLE;
    }

  NodeContainer ssNodes;
```

```
  NodeContainer bsNodes;

  ssNodes.Create (nbSS);
  bsNodes.Create (1);

  WimaxHelper wimax;

  NetDeviceContainer ssDevs, bsDevs;  //instantiate both devices

    Ptr<SimpleOfdmWimaxChannel> channel; // declaring a pointer to channel
  WimaxPhy::Attach (Ptr< WimaxChannel > channel);
  Ptr<WimaxPhy> phy; //declare a pointer to the phyical layer
  phy = CreateObject<SimpleOfdmWimaxPhy> ();

  channel = CreateObject<SimpleOfdmWimaxChannel> (); //set up channels for the
communication
  channel->SetPropagationModel
(SimpleOfdmWimaxChannel::COST231_PROPAGATION);

//input and output of bandwidth using the object phy
  phy->SetChannelBandwidth (50000000);
  phy->GetChannelBandwidth ();
  std::cout << phy->GetChannelBandwidth ()    << std::endl;

  ssDevs = wimax.Install (ssNodes,
                        WimaxHelper::DEVICE_TYPE_SUBSCRIBER_STATION,
                        WimaxHelper::SIMPLE_PHY_TYPE_OFDM,
                        channel,
                        scheduler);
  bsDevs = wimax.Install (bsNodes, WimaxHelper::DEVICE_TYPE_BASE_STATION,
WimaxHelper::SIMPLE_PHY_TYPE_OFDM, channel, scheduler); // BS is constant this
does not move



  Ptr<SubscriberStationNetDevice> ss[nbSS];

  for (int i = 0; i < nbSS; i++)
    {
      ss[i] = ssDevs.Get (i)->GetObject<SubscriberStationNetDevice> ();
      ss[i]->SetModulationType (WimaxPhy::MODULATION_TYPE_QAM16_12); // could
use a switch statement based on modulation required
    }

  Ptr<BaseStationNetDevice> bs;
  bs = bsDevs.Get (0)->GetObject<BaseStationNetDevice> ();
```

```
//set mobility movement
////////////////////////////////////////////////////////// added mobility code start

  Ptr<ConstantPositionMobilityModel> BSPosition;
  Ptr<RandomWaypointMobilityModel> SSPosition[nbSS];
  Ptr<RandomRectanglePositionAllocator> SSPosAllocator[nbSS];
// position of the BS is constant e.g. it does not move

  BSPosition = CreateObject<ConstantPositionMobilityModel> ();

  BSPosition->SetPosition (Vector (1000, 0, 0));
  bsNodes.Get (0)->AggregateObject (BSPosition);
  bsDevs.Add (bsDevs);

//position of the SS mobility 'brownian model'

  for (int i = 0; i < nbSS / 2; i++)
    {
      SSPosition[i] = CreateObject<RandomWaypointMobilityModel> ();
      SSPosAllocator[i] = CreateObject<RandomRectanglePositionAllocator> ();
      SSPosAllocator[i]->SetX (UniformVariable ((i / 40) * 2000, (i /40 + 1) * 2000));
      SSPosAllocator[i]->SetY (UniformVariable ((i / 40) * 2000, (i /40 + 1) * 2000));
      SSPosition[i]->SetAttribute ("PositionAllocator", PointerValue(SSPosAllocator[i]));
      SSPosition[i]->SetAttribute ("Speed", RandomVariableValue (UniformVariable
(10.3, 40.7)));
      SSPosition[i]->SetAttribute ("Pause", RandomVariableValue(ConstantVariable
(0.01)));
      ssNodes.Get (i)->AggregateObject (SSPosition[i]);
    }
///////////////////////////////////////////////////////////////////////////////// added mobility
code end

  MobilityHelper mobility;
  mobility.Install (bsNodes);
  mobility.Install (ssNodes);


///////////////////////////////////////////////////////////// above thre lines is original code


  InternetStackHelper stack;
  stack.Install (bsNodes);
  stack.Install (ssNodes);
```

```
  Ipv4AddressHelper address;
  address.SetBase ("10.1.1.0", "255.255.255.0");

  Ipv4InterfaceContainer SSinterfaces = address.Assign (ssDevs);
  Ipv4InterfaceContainer BSinterface = address.Assign (bsDevs);

 if (verbose)
    {
      wimax.EnableLogComponents ();  // Turn on all wimax logging if true
    }
  /*--------instantiating applications with SS nodes----------------------*/
  UdpServerHelper udpServer[nbSS / 2];
  ApplicationContainer serverApps[nbSS / 2];
  UdpClientHelper udpClient[nbSS / 2];
  ApplicationContainer clientApps[nbSS / 2];

  for (int i = 0; i < nbSS / 2; i++)
    {
      // set server port to 100+(i*10)
      udpServer[i] = UdpServerHelper (100 + (i * 10));
      serverApps[i] = udpServer[i].Install (ssNodes.Get (i));
      serverApps[i].Start (Seconds (0.1)); //start before the client - it then schedules
events
      serverApps[i].Stop (Seconds (duration));

      udpClient[i] = UdpClientHelper (SSinterfaces.GetAddress (i), 100 + (i * 10));
      udpClient[i].SetAttribute ("MaxPackets", UintegerValue (200)); //The maximum
number of packets the application will send
      udpClient[i].SetAttribute ("Interval", TimeValue (Seconds (0.02+0.001))); //The
time to wait between packets
      udpClient[i].SetAttribute ("PacketSize", UintegerValue (1024)); //Size of packets
generated. The minimum packet size is 12 bytes which is the size of the header
carrying the sequence number and the time stamp.

      clientApps[i] = udpClient[i].Install (ssNodes.Get (i + (nbSS / 2)));
      clientApps[i].Start (Seconds (1.5));
      clientApps[i].Stop (Seconds (duration));
    }

  Simulator::Stop (Seconds (duration + 1.1));



  /*
   * Setup 1 transport connections between each SS and the BS, this should be
```

```
determined by the priority
   */
  for (int i = 0; i < nbSS / 2; i++)
    {

      ServiceFlow::SchedulingType x = ServiceFlow::SF_TYPE_BE;
      //ns3::WimaxPhy::SetChannelBandwidth(uint32_t 50000000);

//this is where the DL scheduling starts for the ss
 Throughput(interval);

        switch (i)
          {
          case 0:
            x = ServiceFlow::SF_TYPE_UGS;
            break;
          case 1:
            x = ServiceFlow::SF_TYPE_RTPS;
            break;
          case 2:
            x = ServiceFlow::SF_TYPE_NRTPS;
            break;
          case 3:
            x = ServiceFlow::SF_TYPE_BE;
            break;
          case 4:
            x = ServiceFlow::SF_TYPE_UGS;
            break;
          case 5:
            x = ServiceFlow::SF_TYPE_RTPS;
            break;
          case 6:
            x = ServiceFlow::SF_TYPE_NRTPS;
            break;
               default:
            x = ServiceFlow::SF_TYPE_BE;
          }

            IpcsClassifierRecord DlClassifierBe (Ipv4Address ("0.0.0.0"),
                                                 Ipv4Mask ("0.0.0.0"),
                                                 SSinterfaces.GetAddress (i),
                                                 Ipv4Mask ("255.255.255.255"),
                                                 0,
                                                 65000,
                                                 100 + (i * 10),
                                                 100 + (i * 10),
```

```
                                    17,
                                    1);


      ServiceFlow DlServiceFlowBe = wimax.CreateServiceFlow
(ServiceFlow::SF_DIRECTION_DOWN,
                                           x,          //ServiceFlow::SF_TY
PE_BE is the default,
                                           DlClassifierBe);

//this is where the ul scheduling starts for the ss
    switch (i)
      {
      case 0:
        x = ServiceFlow::SF_TYPE_UGS;
        break;
      case 1:
        x = ServiceFlow::SF_TYPE_UGS;
        break;
      case 2:
        x = ServiceFlow::SF_TYPE_UGS;
        break;
      case 3:
        x = ServiceFlow::SF_TYPE_BE;
        break;
      case 4:
        x = ServiceFlow::SF_TYPE_UGS;
        break;
      case 5:
        x = ServiceFlow::SF_TYPE_RTPS;
        break;
      case 6:
        x = ServiceFlow::SF_TYPE_NRTPS;
      default:
        x = ServiceFlow::SF_TYPE_UGS;
      }

   ss[i]->AddServiceFlow (DlServiceFlowBe);
   IpcsClassifierRecord ulClassifierBe (SSinterfaces.GetAddress (i + (nbSS / 2)),
                                  Ipv4Mask ("255.255.255.255"),
                                  Ipv4Address ("0.0.0.0"),
                                  Ipv4Mask ("0.0.0.0"),
                                  0,
                                  65000,
                                  100 + (i * 10),
                                  100 + (i * 10),
```

```
                                    17,
                                    1);
     ServiceFlow ulServiceFlowBe = wimax.CreateServiceFlow
(ServiceFlow::SF_DIRECTION_UP,
                                                   x, //ServiceFlow::SF_TYPE_RTPS,
                                                   ulClassifierBe);
     ss[i + (nbSS / 2)]->AddServiceFlow (ulServiceFlowBe);



   }


//ReceivedPacket (context, p, addr);


  NS_LOG_INFO ("Starting simulation.....");
  Simulator::Run ();

  for (int i = 0; i < nbSS; i++)
    {
       ss[i] = 0;
       SSPosition[i] = 0;
       SSPosAllocator[i] = 0;

    }
  bs = 0;

  Simulator::Destroy ();
  NS_LOG_INFO ("Done.");

////////////////////////////////////////////////////// gnuplot open

   m_gnuplot.AddDataset (m_output);
   std::ofstream outfile ("throughput.plt");
   m_gnuplot.GenerateOutput (outfile);
   std::cout << "Throughput from Node 2:    " << (BytesRcv * 8 /1000000) /
(duration - 0.5) << " Mbps" << std::endl; // calculation changes hz to mbps as the
bandwidth is
   outfile.close();

//////////////////////////////////////////////////////gnuplot close

  return 0;
}
```

Mobility Models: RandomWalk Data

This is the average of each simulation being run 100 times each.

Random Walk 1 Node

0	110.168	77.1014
2	112.045	76.4109
4	111.428	74.5086
6	112.449	72.7892
8	114.121	73.8864
10	115.89	74.8208
12	114.225	75.9294
14	115.23	74.1999
16	117.101	73.4945
18	119.101	73.4959
20	118.42	75.3765
22	116.474	74.9173
24	114.855	76.0922
26	116.537	77.1747
28	115.342	75.5713
30	115.826	77.5117
32	115.048	75.6692
34	116.932	74.9967
36	117.621	76.874
38	119.477	76.1277
40	117.763	75.0963
42	118.847	73.4153
44	118.42	75.3692
46	119.043	73.4686
48	121.031	73.687
50	121.016	71.6871
52	122.69	70.5924
54	120.692	70.515
56	119.694	68.7817
58	121.693	68.7316
60	122.427	70.5921
62	123.02	72.5021
64	123.466	74.4519
66	121.856	75.6387
68	121.34	77.5712
70	122.823	78.9131
72	124.337	77.6065
74	125.593	79.1635
76	124.018	80.3968
78	125.917	79.767

80	126.837	77.9914
82	124.857	78.2762
84	123.446	76.8597
86	124.408	78.613
88	122.776	79.7692
90	124.494	80.7919
92	123.547	82.5532
94	123.968	80.598
96	122.707	79.0458
98	124.58	79.7454

RandomWalk 2 Nodes								
0ns	0x90f4900	110.168	77.1014		0ns	0x90f4b28	87.6394	90.1827
2000000000ns	0x90f4900	112.045	76.4109		2000000000ns	0x90f4b28	89.6308	89.9981
4000000000ns	0x90f4900	111.428	74.5086		4000000000ns	0x90f4b28	88.5465	91.6786
6000000000ns	0x90f4900	112.449	72.7892		6000000000ns	0x90f4b28	89.9267	93.126
8000000000ns	0x90f4900	114.121	73.8864		8000000000ns	0x90f4b28	91.6543	94.1337
10000000000ns	0x90f4900	115.89	74.8208		10000000000ns	0x90f4b28	92.6767	92.4147
12000000000ns	0x90f4900	114.225	75.9294		12000000000ns	0x90f4b28	93.9238	93.9783
14000000000ns	0x90f4900	115.23	74.1999		14000000000ns	0x90f4b28	92.0323	94.6283
16000000000ns	0x90f4900	117.101	73.4945		16000000000ns	0x90f4b28	90.4858	95.8965
18000000000ns	0x90f4900	119.101	73.4959		18000000000ns	0x90f4b28	89.2127	97.4389
20000000000ns	0x90f4900	118.42	75.3765		20000000000ns	0x90f4b28	87.5511	98.5521
22000000000ns	0x90f4900	116.474	74.9173		22000000000ns	0x90f4b28	86.0185	99.837
24000000000ns	0x90f4900	114.855	76.0922		24000000000ns	0x90f4b28	84.5262	101.169
26000000000ns	0x90f4900	116.537	77.1747		26000000000ns	0x90f4b28	86.4129	100.505
28000000000ns	0x90f4900	115.342	75.5713		28000000000ns	0x90f4b28	85.9407	102.448
30000000000ns	0x90f4900	115.826	77.5117		30000000000ns	0x90f4b28	87.4868	101.18
32000000000ns	0x90f4900	115.048	75.6692		32000000000ns	0x90f4b28	86.01	99.8309
34000000000ns	0x90f4900	116.932	74.9967		34000000000ns	0x90f4b28	85.0489	98.077
36000000000ns	0x90f4900	117.621	76.874		36000000000ns	0x90f4b28	86.5358	99.4146
38000000000ns	0x90f4900	119.477	76.1277		38000000000ns	0x90f4b28	86.5231	97.4146
40000000000ns	0x90f4900	117.763	75.0963		40000000000ns	0x90f4b28	87.1877	95.5283
42000000000ns	0x90f4900	118.847	73.4153		42000000000ns	0x90f4b28	85.1982	95.3233
44000000000ns	0x90f4900	118.42	75.3692		44000000000ns	0x90f4b28	87.0316	94.524
46000000000ns	0x90f4900	119.043	73.4686		46000000000ns	0x90f4b28	88.2287	92.9219
48000000000ns	0x90f4900	121.031	73.687		48000000000ns	0x90f4b28	88.2273	94.9219
50000000000ns	0x90f4900	121.016	71.6871		50000000000ns	0x90f4b28	89.5676	93.4374
52000000000ns	0x90f4900	122.69	70.5924		52000000000ns	0x90f4b28	90.5724	91.7082
54000000000ns	0x90f4900	120.692	70.515		54000000000ns	0x90f4b28	92.5685	91.8326
56000000000ns	0x90f4900	119.694	68.7817		56000000000ns	0x90f4b28	93.5567	93.5715
58000000000ns	0x90f4900	121.693	68.7316		58000000000ns	0x90f4b28	94.969	94.9876
60000000000ns	0x90f4900	122.427	70.5921		60000000000ns	0x90f4b28	96.8301	95.7199
62000000000ns	0x90f4900	123.02	72.5021		62000000000ns	0x90f4b28	98.5143	96.7986
64000000000ns	0x90f4900	123.466	74.4519		64000000000ns	0x90f4b28	100.128	97.9806
66000000000ns	0x90f4900	121.856	75.6387		66000000000ns	0x90f4b28	101.569	99.3671
68000000000ns	0x90f4900	121.34	77.5712		68000000000ns	0x90f4b28	100.81	97.5165
70000000000ns	0x90f4900	122.823	78.9131		70000000000ns	0x90f4b28	102.466	98.6387
72000000000ns	0x90f4900	124.337	77.6065		72000000000ns	0x90f4b28	104.291	97.8215
74000000000ns	0x90f4900	125.593	79.1635		74000000000ns	0x90f4b28	102.393	98.4501
76000000000ns	0x90f4900	124.018	80.3968		76000000000ns	0x90f4b28	104.111	99.4734
78000000000ns	0x90f4900	125.917	79.767		78000000000ns	0x90f4b28	104.707	101.382
80000000000ns	0x90f4900	126.837	77.9914		80000000000ns	0x90f4b28	106.602	100.742
82000000000ns	0x90f4900	124.857	78.2762		82000000000ns	0x90f4b28	107.929	99.2458
84000000000ns	0x90f4900	123.446	76.8597		84000000000ns	0x90f4b28	107.446	97.3051
86000000000ns	0x90f4900	124.408	78.613		86000000000ns	0x90f4b28	105.966	98.6504

88000000000ns	0x90f4900	122.776	79.7692		88000000000ns	0x90f4b28	106.81	96.8374
90000000000ns	0x90f4900	124.494	80.7919		90000000000ns	0x90f4b28	107.603	98.6736
92000000000ns	0x90f4900	123.547	82.5532		92000000000ns	0x90f4b28	109.255	99.8006
94000000000ns	0x90f4900	123.968	80.598		94000000000ns	0x90f4b28	110.986	98.7983
96000000000ns	0x90f4900	122.707	79.0458		96000000000ns	0x90f4b28	109.548	97.4077
98000000000ns	0x90f4900	124.58	79.7454		98000000000ns	0x90f4b28	111.421	98.1103

RandomWalk 4 nodes			
0ns	0x9a88a50	110.168	77.1014
0ns	0x9a88c68	87.6394	90.1827
0ns	0x9a88d50	87.669	100.344
0ns	0x9a88e58	99.6406	100.331
2000000000ns	0x9a88a50	112.045	76.4109
2000000000ns	0x9a88c68	89.6308	89.9981
2000000000ns	0x9a88d50	86.6052	98.6505
2000000000ns	0x9a88e58	98.4713	98.7082
4000000000ns	0x9a88a50	111.428	74.5086
4000000000ns	0x9a88c68	88.5465	91.6786
4000000000ns	0x9a88d50	86.2388	96.6844
4000000000ns	0x9a88e58	96.5646	98.1048
6000000000ns	0x9a88a50	112.449	72.7892
6000000000ns	0x9a88c68	89.9267	93.126
6000000000ns	0x9a88d50	84.5054	97.6819
6000000000ns	0x9a88e58	94.6953	97.3935
8000000000ns	0x9a88a50	114.121	73.8864
8000000000ns	0x9a88c68	91.6543	94.1337
8000000000ns	0x9a88d50	83.9662	99.6079
8000000000ns	0x9a88e58	92.9826	96.3608
10000000000ns	0x9a88a50	115.89	74.8208
10000000000ns	0x9a88c68	92.6767	92.4147
10000000000ns	0x9a88d50	81.9711	99.4687
10000000000ns	0x9a88e58	91.0076	96.0455
12000000000ns	0x9a88a50	114.225	75.9294
12000000000ns	0x9a88c68	93.9238	93.9783
12000000000ns	0x9a88d50	83.6099	98.3223
12000000000ns	0x9a88e58	89.0592	96.4971
14000000000ns	0x9a88a50	115.23	74.1999
14000000000ns	0x9a88c68	92.0323	94.6283
14000000000ns	0x9a88d50	82.4046	96.7264
14000000000ns	0x9a88e58	90.7041	95.3594
16000000000ns	0x9a88a50	117.101	73.4945
16000000000ns	0x9a88c68	90.4858	95.8965
16000000000ns	0x9a88d50	81.4413	98.4791
16000000000ns	0x9a88e58	88.801	95.9744
18000000000ns	0x9a88a50	119.101	73.4959
18000000000ns	0x9a88c68	89.2127	97.4389
18000000000ns	0x9a88d50	79.8464	97.2724
18000000000ns	0x9a88e58	88.0189	97.8152
20000000000ns	0x9a88a50	118.42	75.3765
20000000000ns	0x9a88c68	87.5511	98.5521
20000000000ns	0x9a88d50	77.9197	97.8091
20000000000ns	0x9a88e58	89.155	99.4612

22000000000ns	0x9a88a50	116.474	74.9173
22000000000ns	0x9a88c68	86.0185	99.837
22000000000ns	0x9a88d50	76.7361	96.1969
22000000000ns	0x9a88e58	88.8886	101.443
24000000000ns	0x9a88a50	114.855	76.0922
24000000000ns	0x9a88c68	84.5262	101.169
24000000000ns	0x9a88d50	76.0535	98.0768
24000000000ns	0x9a88e58	89.1231	103.43
26000000000ns	0x9a88a50	116.537	77.1747
26000000000ns	0x9a88c68	86.4129	100.505
26000000000ns	0x9a88d50	74.6298	96.6722
26000000000ns	0x9a88e58	91.1193	103.305
28000000000ns	0x9a88a50	115.342	75.5713
28000000000ns	0x9a88c68	85.9407	102.448
28000000000ns	0x9a88d50	73.6829	98.4339
28000000000ns	0x9a88e58	92.011	101.515
30000000000ns	0x9a88a50	115.826	77.5117
30000000000ns	0x9a88c68	87.4868	101.18
30000000000ns	0x9a88d50	72.9275	100.286
30000000000ns	0x9a88e58	92.803	103.352
32000000000ns	0x9a88a50	115.048	75.6692
32000000000ns	0x9a88c68	86.01	99.8309
32000000000ns	0x9a88d50	73.8672	102.051
32000000000ns	0x9a88e58	94.1985	104.784
34000000000ns	0x9a88a50	116.932	74.9967
34000000000ns	0x9a88c68	85.0489	98.077
34000000000ns	0x9a88d50	72.8329	100.339
34000000000ns	0x9a88e58	92.4983	103.731
36000000000ns	0x9a88a50	117.621	76.874
36000000000ns	0x9a88c68	86.5358	99.4146
36000000000ns	0x9a88d50	73.2656	98.3868
36000000000ns	0x9a88e58	94.4509	104.164
38000000000ns	0x9a88a50	119.477	76.1277
38000000000ns	0x9a88c68	86.5231	97.4146
38000000000ns	0x9a88d50	75.2516	98.1502
38000000000ns	0x9a88e58	92.6802	103.234
40000000000ns	0x9a88a50	117.763	75.0963
40000000000ns	0x9a88c68	87.1877	95.5283
40000000000ns	0x9a88d50	76.0438	96.3138
40000000000ns	0x9a88e58	94.3323	104.361
42000000000ns	0x9a88a50	118.847	73.4153
42000000000ns	0x9a88c68	85.1982	95.3233
42000000000ns	0x9a88d50	74.5016	95.0404
42000000000ns	0x9a88e58	95.2615	106.132
44000000000ns	0x9a88a50	118.42	75.3692

44000000000ns	0x9a88c68	87.0316	94.524
44000000000ns	0x9a88d50	76.33	95.8509
44000000000ns	0x9a88e58	93.4243	106.923
46000000000ns	0x9a88a50	119.043	73.4686
46000000000ns	0x9a88c68	88.2287	92.9219
46000000000ns	0x9a88d50	77.2107	97.6466
46000000000ns	0x9a88e58	94.5729	105.286
48000000000ns	0x9a88a50	121.031	73.687
48000000000ns	0x9a88c68	88.2273	94.9219
48000000000ns	0x9a88d50	77.7669	99.5677
48000000000ns	0x9a88e58	93.7504	103.462
50000000000ns	0x9a88a50	121.016	71.6871
50000000000ns	0x9a88c68	89.5676	93.4374
50000000000ns	0x9a88d50	76.2047	100.816
50000000000ns	0x9a88e58	95.6484	102.832
52000000000ns	0x9a88a50	122.69	70.5924
52000000000ns	0x9a88c68	90.5724	91.7082
52000000000ns	0x9a88d50	77.0681	102.62
52000000000ns	0x9a88e58	93.6599	103.046
54000000000ns	0x9a88a50	120.692	70.515
54000000000ns	0x9a88c68	92.5685	91.8326
54000000000ns	0x9a88d50	76.5334	100.693
54000000000ns	0x9a88e58	94.3106	104.937
56000000000ns	0x9a88a50	119.694	68.7817
56000000000ns	0x9a88c68	93.5567	93.5715
56000000000ns	0x9a88d50	75.7836	98.8391
56000000000ns	0x9a88e58	94.0142	106.915
58000000000ns	0x9a88a50	121.693	68.7316
58000000000ns	0x9a88c68	94.969	94.9876
58000000000ns	0x9a88d50	74.7629	97.1192
58000000000ns	0x9a88e58	95.5779	105.668
60000000000ns	0x9a88a50	122.427	70.5921
60000000000ns	0x9a88c68	96.8301	95.7199
60000000000ns	0x9a88d50	73.7749	98.8581
60000000000ns	0x9a88e58	94.07	104.354
62000000000ns	0x9a88a50	123.02	72.5021
62000000000ns	0x9a88c68	98.5143	96.7986
62000000000ns	0x9a88d50	74.1636	96.8962
62000000000ns	0x9a88e58	92.7117	102.886
64000000000ns	0x9a88a50	123.466	74.4519
64000000000ns	0x9a88c68	100.128	97.9806
64000000000ns	0x9a88d50	74.3131	98.8906
64000000000ns	0x9a88e58	90.7613	103.329
66000000000ns	0x9a88a50	121.856	75.6387
66000000000ns	0x9a88c68	101.569	99.3671

66000000000ns	0x9a88d50	72.3425	98.5492
66000000000ns	0x9a88e58	90.8597	101.331
68000000000ns	0x9a88a50	121.34	77.5712
68000000000ns	0x9a88c68	100.81	97.5165
68000000000ns	0x9a88d50	70.3632	98.2622
68000000000ns	0x9a88e58	92.2552	99.8985
70000000000ns	0x9a88a50	122.823	78.9131
70000000000ns	0x9a88c68	102.466	98.6387
70000000000ns	0x9a88d50	71.6949	96.77
70000000000ns	0x9a88e58	94.186	99.377
72000000000ns	0x9a88a50	124.337	77.6065
72000000000ns	0x9a88c68	104.291	97.8215
72000000000ns	0x9a88d50	69.7016	96.9333
72000000000ns	0x9a88e58	93.0395	97.7382
74000000000ns	0x9a88a50	125.593	79.1635
74000000000ns	0x9a88c68	102.393	98.4501
74000000000ns	0x9a88d50	71.1416	95.5454
74000000000ns	0x9a88e58	93.5916	99.6605
76000000000ns	0x9a88a50	124.018	80.3968
76000000000ns	0x9a88c68	104.111	99.4734
76000000000ns	0x9a88d50	73.0418	96.1692
76000000000ns	0x9a88e58	94.8324	101.229
78000000000ns	0x9a88a50	125.917	79.767
78000000000ns	0x9a88c68	104.707	101.382
78000000000ns	0x9a88d50	73.6946	94.2787
78000000000ns	0x9a88e58	95.5201	103.107
80000000000ns	0x9a88a50	126.837	77.9914
80000000000ns	0x9a88c68	106.602	100.742
80000000000ns	0x9a88d50	73.0781	92.3761
80000000000ns	0x9a88e58	94.1763	104.588
82000000000ns	0x9a88a50	124.857	78.2762
82000000000ns	0x9a88c68	107.929	99.2458
82000000000ns	0x9a88d50	73.7405	90.489
82000000000ns	0x9a88e58	96.0407	105.312
84000000000ns	0x9a88a50	123.446	76.8597
84000000000ns	0x9a88c68	107.446	97.3051
84000000000ns	0x9a88d50	74.6237	88.6946
84000000000ns	0x9a88e58	94.167	106.012
86000000000ns	0x9a88a50	124.408	78.613
86000000000ns	0x9a88c68	105.966	98.6504
86000000000ns	0x9a88d50	74.1615	86.7487
86000000000ns	0x9a88e58	92.5629	107.206
88000000000ns	0x9a88a50	122.776	79.7692
88000000000ns	0x9a88c68	106.81	96.8374
88000000000ns	0x9a88d50	76.0859	87.2934

88000000000ns	0x9a88e58	94.0546	105.874
90000000000ns	0x9a88a50	124.494	80.7919
90000000000ns	0x9a88c68	107.603	98.6736
90000000000ns	0x9a88d50	78.0822	87.4151
90000000000ns	0x9a88e58	95.5215	107.233
92000000000ns	0x9a88a50	123.547	82.5532
92000000000ns	0x9a88c68	109.255	99.8006
92000000000ns	0x9a88d50	76.8866	85.8118
92000000000ns	0x9a88e58	96.2265	105.362
94000000000ns	0x9a88a50	123.968	80.598
94000000000ns	0x9a88c68	110.986	98.7983
94000000000ns	0x9a88d50	74.9784	86.4105
94000000000ns	0x9a88e58	97.4202	106.966
96000000000ns	0x9a88a50	122.707	79.0458
96000000000ns	0x9a88c68	109.548	97.4077
96000000000ns	0x9a88d50	76.9693	86.6003
96000000000ns	0x9a88e58	97.8379	108.922
98000000000ns	0x9a88a50	124.58	79.7454
98000000000ns	0x9a88c68	111.421	98.1103
98000000000ns	0x9a88d50	78.1708	88.1992
98000000000ns	0x9a88e58	95.8653	108.593

Random Walk 8 Nodes			
0ns	0x86a0d18	110.168	77.1014
0ns	0x86a0f30	87.6394	90.1827
0ns	0x86a1018	87.669	100.344
0ns	0x86a1120	99.6406	100.331
0ns	0x86a1248	91.0151	109.682
0ns	0x86a1370	109.585	74.1875
0ns	0x86a1498	115.291	111.24
0ns	0x86a15c0	83.9076	90.5132
2000000000ns	0x86a0d18	112.045	76.4109
2000000000ns	0x86a0f30	89.6308	89.9981
2000000000ns	0x86a1018	86.6052	98.6505
2000000000ns	0x86a1120	98.4713	98.7082
2000000000ns	0x86a1248	90.5364	107.74
2000000000ns	0x86a1370	111.458	74.8884
2000000000ns	0x86a1498	116.199	113.022
2000000000ns	0x86a15c0	82.2414	89.4069
4000000000ns	0x86a0d18	111.428	74.5086
4000000000ns	0x86a0f30	88.5465	91.6786
4000000000ns	0x86a1018	86.2388	96.6844
4000000000ns	0x86a1120	96.5646	98.1048
4000000000ns	0x86a1248	88.5668	107.392
4000000000ns	0x86a1370	111.276	76.88
4000000000ns	0x86a1498	114.328	112.316
4000000000ns	0x86a15c0	82.2021	91.4065
6000000000ns	0x86a0d18	112.449	72.7892
6000000000ns	0x86a0f30	89.9267	93.126
6000000000ns	0x86a1018	84.5054	97.6819
6000000000ns	0x86a1120	94.6953	97.3935
6000000000ns	0x86a1248	88.6258	109.392
6000000000ns	0x86a1370	112.856	78.1061
6000000000ns	0x86a1498	113.093	113.889
6000000000ns	0x86a15c0	82.6872	93.3468
8000000000ns	0x86a0d18	114.121	73.8864
8000000000ns	0x86a0f30	91.6543	94.1337
8000000000ns	0x86a1018	83.9662	99.6079
8000000000ns	0x86a1120	92.9826	96.3608
8000000000ns	0x86a1248	88.5887	107.392
8000000000ns	0x86a1370	112.832	80.106
8000000000ns	0x86a1498	112.449	115.783
8000000000ns	0x86a15c0	80.8606	94.1612
10000000000ns	0x86a0d18	115.89	74.8208
10000000000ns	0x86a0f30	92.6767	92.4147
10000000000ns	0x86a1018	81.9711	99.4687

10000000000ns	0x86a1120	91.0076	96.0455
10000000000ns	0x86a1248	89.2687	109.273
10000000000ns	0x86a1370	114.724	80.7562
10000000000ns	0x86a1498	111.178	117.327
10000000000ns	0x86a15c0	80.8517	92.1612
12000000000ns	0x86a0d18	114.225	75.9294
12000000000ns	0x86a0f30	93.9238	93.9783
12000000000ns	0x86a1018	83.6099	98.3223
12000000000ns	0x86a1120	89.0592	96.4971
12000000000ns	0x86a1248	90.0233	107.421
12000000000ns	0x86a1370	113.917	82.5861
12000000000ns	0x86a1498	111.39	119.316
12000000000ns	0x86a15c0	80.2643	94.073
14000000000ns	0x86a0d18	115.23	74.1999
14000000000ns	0x86a0f30	92.0323	94.6283
14000000000ns	0x86a1018	82.4046	96.7264
14000000000ns	0x86a1120	90.7041	95.3594
14000000000ns	0x86a1248	89.6586	109.387
14000000000ns	0x86a1370	114.134	80.598
14000000000ns	0x86a1498	110.161	120.893
14000000000ns	0x86a15c0	82.0559	93.1842
16000000000ns	0x86a0d18	117.101	73.4945
16000000000ns	0x86a0f30	90.4858	95.8965
16000000000ns	0x86a1018	81.4413	98.4791
16000000000ns	0x86a1120	88.801	95.9744
16000000000ns	0x86a1248	87.6591	109.431
16000000000ns	0x86a1370	115.539	79.174
16000000000ns	0x86a1498	111.17	122.62
16000000000ns	0x86a15c0	83.5204	94.5463
18000000000ns	0x86a0d18	119.101	73.4959
18000000000ns	0x86a0f30	89.2127	97.4389
18000000000ns	0x86a1018	79.8464	97.2724
18000000000ns	0x86a1120	88.0189	97.8152
18000000000ns	0x86a1248	88.7037	107.726
18000000000ns	0x86a1370	113.552	79.4064
18000000000ns	0x86a1498	109.171	122.686
18000000000ns	0x86a15c0	84.4978	92.8014
20000000000ns	0x86a0d18	118.42	75.3765
20000000000ns	0x86a0f30	87.5511	98.5521
20000000000ns	0x86a1018	77.9197	97.8091
20000000000ns	0x86a1120	89.155	99.4612
20000000000ns	0x86a1248	90.6205	108.296
20000000000ns	0x86a1370	111.86	78.3396
20000000000ns	0x86a1498	108.145	124.403

20000000000ns	0x86a15c0	83.1817	91.2954
22000000000ns	0x86a0d18	116.474	74.9173
22000000000ns	0x86a0f30	86.0185	99.837
22000000000ns	0x86a1018	76.7361	96.1969
22000000000ns	0x86a1120	88.8886	101.443
22000000000ns	0x86a1248	90.943	106.323
22000000000ns	0x86a1370	112.557	76.4648
22000000000ns	0x86a1498	109.127	122.66
22000000000ns	0x86a15c0	82.2633	93.072
24000000000ns	0x86a0d18	114.855	76.0922
24000000000ns	0x86a0f30	84.5262	101.169
24000000000ns	0x86a1018	76.0535	98.0768
24000000000ns	0x86a1120	89.1231	103.43
24000000000ns	0x86a1248	92.9355	106.496
24000000000ns	0x86a1370	112.667	78.4618
24000000000ns	0x86a1498	110.6	124.013
24000000000ns	0x86a15c0	82.2556	91.072
26000000000ns	0x86a0d18	116.537	77.1747
26000000000ns	0x86a0f30	86.4129	100.505
26000000000ns	0x86a1018	74.6298	96.6722
26000000000ns	0x86a1120	91.1193	103.305
26000000000ns	0x86a1248	92.3698	108.414
26000000000ns	0x86a1370	113.574	80.2443
26000000000ns	0x86a1498	109.287	125.522
26000000000ns	0x86a15c0	83.205	92.8323
28000000000ns	0x86a0d18	115.342	75.5713
28000000000ns	0x86a0f30	85.9407	102.448
28000000000ns	0x86a1018	73.6829	98.4339
28000000000ns	0x86a1120	92.011	101.515
28000000000ns	0x86a1248	93.5651	106.811
28000000000ns	0x86a1370	115.368	81.1291
28000000000ns	0x86a1498	111.287	125.53
28000000000ns	0x86a15c0	82.2419	91.0795
30000000000ns	0x86a0d18	115.826	77.5117
30000000000ns	0x86a0f30	87.4868	101.18
30000000000ns	0x86a1018	72.9275	100.286
30000000000ns	0x86a1120	92.803	103.352
30000000000ns	0x86a1248	92.6743	105.02
30000000000ns	0x86a1370	115.746	83.093
30000000000ns	0x86a1498	110.672	123.627
30000000000ns	0x86a15c0	83.3696	92.7312
32000000000ns	0x86a0d18	115.048	75.6692
32000000000ns	0x86a0f30	86.01	99.8309
32000000000ns	0x86a1018	73.8672	102.051

32000000000ns	0x86a1120	94.1985	104.784
32000000000ns	0x86a1248	94.309	103.868
32000000000ns	0x86a1370	117.618	82.3887
32000000000ns	0x86a1498	109.985	121.749
32000000000ns	0x86a15c0	81.5556	93.5734
34000000000ns	0x86a0d18	116.932	74.9967
34000000000ns	0x86a0f30	85.0489	98.077
34000000000ns	0x86a1018	72.8329	100.339
34000000000ns	0x86a1120	92.4983	103.731
34000000000ns	0x86a1248	96.2645	104.287
34000000000ns	0x86a1370	119.507	83.0463
34000000000ns	0x86a1498	109.163	123.572
34000000000ns	0x86a15c0	83.5368	93.3001
36000000000ns	0x86a0d18	117.621	76.874
36000000000ns	0x86a0f30	86.5358	99.4146
36000000000ns	0x86a1018	73.2656	98.3868
36000000000ns	0x86a1120	94.4509	104.164
36000000000ns	0x86a1248	97.8593	103.08
36000000000ns	0x86a1370	121.431	82.4995
36000000000ns	0x86a1498	107.177	123.33
36000000000ns	0x86a15c0	85.3117	92.3782
38000000000ns	0x86a0d18	119.477	76.1277
38000000000ns	0x86a0f30	86.5231	97.4146
38000000000ns	0x86a1018	75.2516	98.1502
38000000000ns	0x86a1120	92.6802	103.234
38000000000ns	0x86a1248	99.8593	103.077
38000000000ns	0x86a1370	120.629	80.667
38000000000ns	0x86a1498	107.024	121.336
38000000000ns	0x86a15c0	84.396	90.6001
40000000000ns	0x86a0d18	117.763	75.0963
40000000000ns	0x86a0f30	87.1877	95.5283
40000000000ns	0x86a1018	76.0438	96.3138
40000000000ns	0x86a1120	94.3323	104.361
40000000000ns	0x86a1248	101.553	102.014
40000000000ns	0x86a1370	122.078	82.046
40000000000ns	0x86a1498	106.597	119.383
40000000000ns	0x86a15c0	85.3417	88.8378
42000000000ns	0x86a0d18	118.847	73.4153
42000000000ns	0x86a0f30	85.1982	95.3233
42000000000ns	0x86a1018	74.5016	95.0404
42000000000ns	0x86a1120	95.2615	106.132
42000000000ns	0x86a1248	102.303	100.16
42000000000ns	0x86a1370	120.167	82.6363
42000000000ns	0x86a1498	108.259	118.271

42000000000ns	0x86a15c0	85.3224	90.8377
44000000000ns	0x86a0d18	118.42	75.3692
44000000000ns	0x86a0f30	87.0316	94.524
44000000000ns	0x86a1018	76.33	95.8509
44000000000ns	0x86a1120	93.4243	106.923
44000000000ns	0x86a1248	102.236	102.159
44000000000ns	0x86a1370	118.951	81.048
44000000000ns	0x86a1498	108.269	120.271
44000000000ns	0x86a15c0	85.1802	92.8326
46000000000ns	0x86a0d18	119.043	73.4686
46000000000ns	0x86a0f30	88.2287	92.9219
46000000000ns	0x86a1018	77.2107	97.6466
46000000000ns	0x86a1120	94.5729	105.286
46000000000ns	0x86a1248	102.327	100.161
46000000000ns	0x86a1370	120.181	82.6254
46000000000ns	0x86a1498	107.96	122.247
46000000000ns	0x86a15c0	83.4861	91.7696
48000000000ns	0x86a0d18	121.031	73.687
48000000000ns	0x86a0f30	88.2273	94.9219
48000000000ns	0x86a1018	77.7669	99.5677
48000000000ns	0x86a1120	93.7504	103.462
48000000000ns	0x86a1248	103.888	98.9107
48000000000ns	0x86a1370	121.184	84.3557
48000000000ns	0x86a1498	109.749	121.353
48000000000ns	0x86a15c0	83.8808	89.8089
50000000000ns	0x86a0d18	121.016	71.6871
50000000000ns	0x86a0f30	89.5676	93.4374
50000000000ns	0x86a1018	76.2047	100.816
50000000000ns	0x86a1120	95.6484	102.832
50000000000ns	0x86a1248	105.116	97.3324
50000000000ns	0x86a1370	119.715	82.9979
50000000000ns	0x86a1498	108.511	122.924
50000000000ns	0x86a15c0	82.1388	90.7914
52000000000ns	0x86a0d18	122.69	70.5924
52000000000ns	0x86a0f30	90.5724	91.7082
52000000000ns	0x86a1018	77.0681	102.62
52000000000ns	0x86a1120	93.6599	103.046
52000000000ns	0x86a1248	105.757	99.227
52000000000ns	0x86a1370	118.548	81.3742
52000000000ns	0x86a1498	109.968	121.554
52000000000ns	0x86a15c0	82.7304	88.881
54000000000ns	0x86a0d18	120.692	70.515
54000000000ns	0x86a0f30	92.5685	91.8326
54000000000ns	0x86a1018	76.5334	100.693

54000000000ns	0x86a1120	94.3106	104.937
54000000000ns	0x86a1248	105.677	101.225
54000000000ns	0x86a1370	117.175	79.9201
54000000000ns	0x86a1498	107.973	121.685
54000000000ns	0x86a15c0	84.4427	89.9146
56000000000ns	0x86a0d18	119.694	68.7817
56000000000ns	0x86a0f30	93.5567	93.5715
56000000000ns	0x86a1018	75.7836	98.8391
56000000000ns	0x86a1120	94.0142	106.915
56000000000ns	0x86a1248	104.027	102.356
56000000000ns	0x86a1370	117.429	81.9038
56000000000ns	0x86a1498	107.117	119.877
56000000000ns	0x86a15c0	85.1875	91.7707
58000000000ns	0x86a0d18	121.693	68.7316
58000000000ns	0x86a0f30	94.969	94.9876
58000000000ns	0x86a1018	74.7629	97.1192
58000000000ns	0x86a1120	95.5779	105.668
58000000000ns	0x86a1248	102.936	104.032
58000000000ns	0x86a1370	119.414	82.1495
58000000000ns	0x86a1498	108.439	118.376
58000000000ns	0x86a15c0	85.42	89.7842
60000000000ns	0x86a0d18	122.427	70.5921
60000000000ns	0x86a0f30	96.8301	95.7199
60000000000ns	0x86a1018	73.7749	98.8581
60000000000ns	0x86a1120	94.07	104.354
60000000000ns	0x86a1248	100.954	104.301
60000000000ns	0x86a1370	120.18	83.9972
60000000000ns	0x86a1498	108.204	116.39
60000000000ns	0x86a15c0	85.7188	91.7618
62000000000ns	0x86a0d18	123.02	72.5021
62000000000ns	0x86a0f30	98.5143	96.7986
62000000000ns	0x86a1018	74.1636	96.8962
62000000000ns	0x86a1120	92.7117	102.886
62000000000ns	0x86a1248	102.954	104.277
62000000000ns	0x86a1370	120.883	82.1247
62000000000ns	0x86a1498	109.155	114.631
62000000000ns	0x86a15c0	85.1097	93.6668
64000000000ns	0x86a0d18	123.466	74.4519
64000000000ns	0x86a0f30	100.128	97.9806
64000000000ns	0x86a1018	74.3131	98.8906
64000000000ns	0x86a1120	90.7613	103.329
64000000000ns	0x86a1248	100.954	104.336
64000000000ns	0x86a1370	122.209	80.6275
64000000000ns	0x86a1498	109.61	112.683

64000000000ns	0x86a15c0	83.1796	93.1426
66000000000ns	0x86a0d18	121.856	75.6387
66000000000ns	0x86a0f30	101.569	99.3671
66000000000ns	0x86a1018	72.3425	98.5492
66000000000ns	0x86a1120	90.8597	101.331
66000000000ns	0x86a1248	100.1	102.527
66000000000ns	0x86a1370	123.402	79.0222
66000000000ns	0x86a1498	107.619	112.881
66000000000ns	0x86a15c0	85.1471	92.7834
68000000000ns	0x86a0d18	121.34	77.5712
68000000000ns	0x86a0f30	100.81	97.5165
68000000000ns	0x86a1018	70.3632	98.2622
68000000000ns	0x86a1120	92.2552	99.8985
68000000000ns	0x86a1248	98.1753	101.983
68000000000ns	0x86a1370	124.604	80.62
68000000000ns	0x86a1498	109.239	114.055
68000000000ns	0x86a15c0	84.9884	94.7771
70000000000ns	0x86a0d18	122.823	78.9131
70000000000ns	0x86a0f30	102.466	98.6387
70000000000ns	0x86a1018	71.6949	96.77
70000000000ns	0x86a1120	94.186	99.377
70000000000ns	0x86a1248	96.8344	103.467
70000000000ns	0x86a1370	126.542	80.1229
70000000000ns	0x86a1498	107.348	114.707
70000000000ns	0x86a15c0	85.3504	96.7441
72000000000ns	0x86a0d18	124.337	77.6065
72000000000ns	0x86a0f30	104.291	97.8215
72000000000ns	0x86a1018	69.7016	96.9333
72000000000ns	0x86a1120	93.0395	97.7382
72000000000ns	0x86a1248	98.6966	104.197
72000000000ns	0x86a1370	126.548	78.1229
72000000000ns	0x86a1498	105.349	114.683
72000000000ns	0x86a15c0	83.6167	95.7469
74000000000ns	0x86a0d18	125.593	79.1635
74000000000ns	0x86a0f30	102.393	98.4501
74000000000ns	0x86a1018	71.1416	95.5454
74000000000ns	0x86a1120	93.5916	99.6605
74000000000ns	0x86a1248	96.8625	103.399
74000000000ns	0x86a1370	125.985	76.2038
74000000000ns	0x86a1498	106.78	116.079
74000000000ns	0x86a15c0	83.5861	93.7472
76000000000ns	0x86a0d18	124.018	80.3968
76000000000ns	0x86a0f30	104.111	99.4734
76000000000ns	0x86a1018	73.0418	96.1692

76000000000ns	0x86a1120	94.8324	101.229
76000000000ns	0x86a1248	97.1542	105.378
76000000000ns	0x86a1370	124.307	75.1148
76000000000ns	0x86a1498	108.07	114.551
76000000000ns	0x86a15c0	82.9402	91.8543
78000000000ns	0x86a0d18	125.917	79.767
78000000000ns	0x86a0f30	104.707	101.382
78000000000ns	0x86a1018	73.6946	94.2787
78000000000ns	0x86a1120	95.5201	103.107
78000000000ns	0x86a1248	98.405	106.939
78000000000ns	0x86a1370	125.215	76.8968
78000000000ns	0x86a1498	109.971	115.171
78000000000ns	0x86a15c0	84.305	93.3163
80000000000ns	0x86a0d18	126.837	77.9914
80000000000ns	0x86a0f30	106.602	100.742
80000000000ns	0x86a1018	73.0781	92.3761
80000000000ns	0x86a1120	94.1763	104.588
80000000000ns	0x86a1248	99.0685	108.825
80000000000ns	0x86a1370	123.219	76.7719
80000000000ns	0x86a1498	111.596	114.004
80000000000ns	0x86a15c0	86.2423	93.813
82000000000ns	0x86a0d18	124.857	78.2762
82000000000ns	0x86a0f30	107.929	99.2458
82000000000ns	0x86a1018	73.7405	90.489
82000000000ns	0x86a1120	96.0407	105.312
82000000000ns	0x86a1248	97.8416	107.246
82000000000ns	0x86a1370	121.355	77.4957
82000000000ns	0x86a1498	113.477	113.324
82000000000ns	0x86a15c0	84.2439	93.734
84000000000ns	0x86a0d18	123.446	76.8597
84000000000ns	0x86a0f30	107.446	97.3051
84000000000ns	0x86a1018	74.6237	88.6946
84000000000ns	0x86a1120	94.167	106.012
84000000000ns	0x86a1248	97.3032	105.32
84000000000ns	0x86a1370	121.612	79.4791
84000000000ns	0x86a1498	112.209	114.871
84000000000ns	0x86a15c0	86.0761	92.9323
86000000000ns	0x86a0d18	124.408	78.613
86000000000ns	0x86a0f30	105.966	98.6504
86000000000ns	0x86a1018	74.1615	86.7487
86000000000ns	0x86a1120	92.5629	107.206
86000000000ns	0x86a1248	95.9788	103.821
86000000000ns	0x86a1370	123.501	80.1345
86000000000ns	0x86a1498	114.136	115.407

86000000000ns	0x86a15c0	87.1874	94.5951
88000000000ns	0x86a0d18	122.776	79.7692
88000000000ns	0x86a0f30	106.81	96.8374
88000000000ns	0x86a1018	76.0859	87.2934
88000000000ns	0x86a1120	94.0546	105.874
88000000000ns	0x86a1248	96.6478	101.936
88000000000ns	0x86a1370	124.802	81.6539
88000000000ns	0x86a1498	116.023	114.745
88000000000ns	0x86a15c0	86.149	92.8858
90000000000ns	0x86a0d18	124.494	80.7919
90000000000ns	0x86a0f30	107.603	98.6736
90000000000ns	0x86a1018	78.0822	87.4151
90000000000ns	0x86a1120	95.5215	107.233
90000000000ns	0x86a1248	94.7908	101.194
90000000000ns	0x86a1370	123.607	83.2582
90000000000ns	0x86a1498	118.002	115.033
90000000000ns	0x86a15c0	84.2561	93.5313
92000000000ns	0x86a0d18	123.547	82.5532
92000000000ns	0x86a0f30	109.255	99.8006
92000000000ns	0x86a1018	76.8866	85.8118
92000000000ns	0x86a1120	96.2265	105.362
92000000000ns	0x86a1248	92.7968	101.039
92000000000ns	0x86a1370	123.259	81.2888
92000000000ns	0x86a1498	116.217	115.935
92000000000ns	0x86a15c0	82.3924	94.2571
94000000000ns	0x86a0d18	123.968	80.598
94000000000ns	0x86a0f30	110.986	98.7983
94000000000ns	0x86a1018	74.9784	86.4105
94000000000ns	0x86a1120	97.4202	106.966
94000000000ns	0x86a1248	92.4033	99.0782
94000000000ns	0x86a1370	122.999	83.2719
94000000000ns	0x86a1498	118.05	115.135
94000000000ns	0x86a15c0	84.0871	95.3192
96000000000ns	0x86a0d18	122.707	79.0458
96000000000ns	0x86a0f30	109.548	97.4077
96000000000ns	0x86a1018	76.9693	86.6003
96000000000ns	0x86a1120	97.8379	108.922
96000000000ns	0x86a1248	90.4609	99.5548
96000000000ns	0x86a1370	123.209	85.2608
96000000000ns	0x86a1498	116.543	113.82
96000000000ns	0x86a15c0	82.908	93.7038
98000000000ns	0x86a0d18	124.58	79.7454
98000000000ns	0x86a0f30	111.421	98.1103
98000000000ns	0x86a1018	78.1708	88.1992

98000000000ns	0x86a1120	95.8653	108.593
98000000000ns	0x86a1248	92.3069	100.324
98000000000ns	0x86a1370	123.401	87.2516
98000000000ns	0x86a1498	118.541	113.912
98000000000ns	0x86a15c0	84.5431	94.8555

Random Walk 32 Nodes			
0ns	0x9a96da8	110.168	77.1014
0ns	0x9a96f98	87.6394	90.1827
0ns	0x9a97080	87.669	100.344
0ns	0x9a97188	99.6406	100.331
0ns	0x9a972b0	91.0151	109.682
0ns	0x9a973d8	109.585	74.1875
0ns	0x9a97500	115.291	111.24
0ns	0x9a97628	83.9076	90.5132
0ns	0x9a97750	107.599	96.9366
0ns	0x9a97878	97.4694	97.4346
0ns	0x9a979a0	110.706	77.9455
0ns	0x9a97ac8	82.5345	93.1799
0ns	0x9a97bf0	100.697	125.648
0ns	0x9a97d18	107.035	88.3781
0ns	0x9a97e58	120.32	102.556
0ns	0x9a97f98	109.526	84.6999
0ns	0x9a980d8	118.089	93.3539
0ns	0x9a98218	115.048	74.6691
0ns	0x9a98358	122.384	109.14
0ns	0x9a98498	82.6805	98.4261
0ns	0x9a985d8	105.061	127.967
0ns	0x9a98718	90.0192	115.535
0ns	0x9a98858	95.973	103.959
0ns	0x9a98998	82.0521	120.381
0ns	0x9a98ad8	100.674	99.82
0ns	0x9a98c18	81.9109	101.373
0ns	0x9a98d58	98.9215	101.447
0ns	0x9a98e98	95.6565	112.542
0ns	0x9a98fd8	102.706	98.557
0ns	0x9a99118	92.0103	75.062
0ns	0x9a99258	90.4964	101.624
0ns	0x9a99398	92.2366	123.865
2000000000ns	0x9a96da8	112.045	76.4109
2000000000ns	0x9a96f98	89.6308	89.9981
2000000000ns	0x9a97080	86.6052	98.6505
2000000000ns	0x9a97188	98.4713	98.7082
2000000000ns	0x9a972b0	90.5364	107.74
2000000000ns	0x9a973d8	111.458	74.8884
2000000000ns	0x9a97500	116.199	113.022
2000000000ns	0x9a97628	82.2414	89.4069
2000000000ns	0x9a97750	106.818	98.7779
2000000000ns	0x9a97878	97.9522	95.4938
2000000000ns	0x9a979a0	108.951	78.906
2000000000ns	0x9a97ac8	83.8851	91.7048

2000000000ns	0x9a97bf0	102.606	126.244
2000000000ns	0x9a97d18	105.239	89.2583
2000000000ns	0x9a97e58	122.317	102.447
2000000000ns	0x9a97f98	107.878	85.8338
2000000000ns	0x9a980d8	119.538	91.9758
2000000000ns	0x9a98218	116.753	75.715
2000000000ns	0x9a98358	124.346	109.525
2000000000ns	0x9a98498	81.7929	100.218
2000000000ns	0x9a985d8	106.001	129.733
2000000000ns	0x9a98718	88.679	114.05
2000000000ns	0x9a98858	94.8517	102.303
2000000000ns	0x9a98998	80.0605	120.197
2000000000ns	0x9a98ad8	100.036	97.9245
2000000000ns	0x9a98c18	83.6037	100.308
2000000000ns	0x9a98d58	98.1459	103.291
2000000000ns	0x9a98e98	95.7769	110.545
2000000000ns	0x9a98fd8	100.706	98.5701
2000000000ns	0x9a99118	90.0137	75.1769
2000000000ns	0x9a99258	89.5389	103.38
2000000000ns	0x9a99398	91.2295	122.137
4000000000ns	0x9a96da8	111.428	74.5086
4000000000ns	0x9a96f98	88.5465	91.6786
4000000000ns	0x9a97080	86.2388	96.6844
4000000000ns	0x9a97188	96.5646	98.1048
4000000000ns	0x9a972b0	88.5668	107.392
4000000000ns	0x9a973d8	111.276	76.88
4000000000ns	0x9a97500	114.328	112.316
4000000000ns	0x9a97628	82.2021	91.4065
4000000000ns	0x9a97750	106.805	100.778
4000000000ns	0x9a97878	99.0916	97.1375
4000000000ns	0x9a979a0	109.187	76.92
4000000000ns	0x9a97ac8	85.8324	91.2488
4000000000ns	0x9a97bf0	100.946	127.358
4000000000ns	0x9a97d18	106.284	90.9637
4000000000ns	0x9a97e58	123.188	104.248
4000000000ns	0x9a97f98	105.904	86.1557
4000000000ns	0x9a980d8	121.161	90.8064
4000000000ns	0x9a98218	116.35	73.7559
4000000000ns	0x9a98358	126.068	110.542
4000000000ns	0x9a98498	83.3603	98.9761
4000000000ns	0x9a985d8	104.284	130.758
4000000000ns	0x9a98718	86.7192	113.651
4000000000ns	0x9a98858	96.8069	102.724
4000000000ns	0x9a98998	78.0607	120.225
4000000000ns	0x9a98ad8	98.5072	96.6354

4000000000ns	0x9a98c18	81.7086	99.669
4000000000ns	0x9a98d58	96.8339	104.8
4000000000ns	0x9a98e98	97.5402	109.602
4000000000ns	0x9a98fd8	102.29	97.3494
4000000000ns	0x9a99118	88.0695	75.6462
4000000000ns	0x9a99258	91.5192	103.1
4000000000ns	0x9a99398	90.7973	124.09
6000000000ns	0x9a96da8	112.449	72.7892
6000000000ns	0x9a96f98	89.9267	93.126
6000000000ns	0x9a97080	84.5054	97.6819
6000000000ns	0x9a97188	94.6953	97.3935
6000000000ns	0x9a972b0	88.6258	109.392
6000000000ns	0x9a973d8	112.856	78.1061
6000000000ns	0x9a97500	113.093	113.889
6000000000ns	0x9a97628	82.6872	93.3468
6000000000ns	0x9a97750	107.947	102.42
6000000000ns	0x9a97878	100.29	98.7383
6000000000ns	0x9a979a0	107.8	78.3603
6000000000ns	0x9a97ac8	86.6982	89.446
6000000000ns	0x9a97bf0	100.547	129.318
6000000000ns	0x9a97d18	107.658	92.4171
6000000000ns	0x9a97e58	124.028	102.432
6000000000ns	0x9a97f98	107.193	84.6262
6000000000ns	0x9a980d8	120.832	92.7792
6000000000ns	0x9a98218	114.369	74.0309
6000000000ns	0x9a98358	128.018	110.986
6000000000ns	0x9a98498	82.6544	100.847
6000000000ns	0x9a985d8	102.572	129.724
6000000000ns	0x9a98718	85.8955	111.829
6000000000ns	0x9a98858	96.3216	104.664
6000000000ns	0x9a98998	76.7454	121.732
6000000000ns	0x9a98ad8	97.5266	94.8923
6000000000ns	0x9a98c18	80.2237	101.009
6000000000ns	0x9a98d58	98.4076	106.034
6000000000ns	0x9a98e98	95.5424	109.697
6000000000ns	0x9a98fd8	103.995	98.3948
6000000000ns	0x9a99118	89.099	73.9315
6000000000ns	0x9a99258	89.5859	102.588
6000000000ns	0x9a99398	89.975	125.913
8000000000ns	0x9a96da8	114.121	73.8864
8000000000ns	0x9a96f98	91.6543	94.1337
8000000000ns	0x9a97080	83.9662	99.6079
8000000000ns	0x9a97188	92.9826	96.3608
8000000000ns	0x9a972b0	88.5887	107.392
8000000000ns	0x9a973d8	112.832	80.106

8000000000ns	0x9a97500	112.449	115.783
8000000000ns	0x9a97628	80.8606	94.1612
8000000000ns	0x9a97750	106.515	103.816
8000000000ns	0x9a97878	101.117	96.9172
8000000000ns	0x9a979a0	105.882	78.9275
8000000000ns	0x9a97ac8	85.3423	87.9757
8000000000ns	0x9a97bf0	102.46	128.735
8000000000ns	0x9a97d18	109.491	93.2155
8000000000ns	0x9a97e58	123.671	100.464
8000000000ns	0x9a97f98	108.092	86.4126
8000000000ns	0x9a980d8	119.29	91.5056
8000000000ns	0x9a98218	116.361	74.2086
8000000000ns	0x9a98358	127.936	108.988
8000000000ns	0x9a98498	81.1947	102.215
8000000000ns	0x9a985d8	100.796	128.804
8000000000ns	0x9a98718	85.3726	109.898
8000000000ns	0x9a98858	98.1749	105.416
8000000000ns	0x9a98998	77.0895	119.762
8000000000ns	0x9a98ad8	96.609	96.6693
8000000000ns	0x9a98c18	81.0754	99.1992
8000000000ns	0x9a98d58	97.3043	104.366
8000000000ns	0x9a98e98	96.2664	107.833
8000000000ns	0x9a98fd8	103.19	100.226
8000000000ns	0x9a99118	91.0656	73.5677
8000000000ns	0x9a99258	89.7817	104.578
8000000000ns	0x9a99398	88.9723	124.182
10000000000ns	0x9a96da8	115.89	74.8208
10000000000ns	0x9a96f98	92.6767	92.4147
10000000000ns	0x9a97080	81.9711	99.4687
10000000000ns	0x9a97188	91.0076	96.0455
10000000000ns	0x9a972b0	89.2687	109.273
10000000000ns	0x9a973d8	114.724	80.7562
10000000000ns	0x9a97500	111.178	117.327
10000000000ns	0x9a97628	80.8517	92.1612
10000000000ns	0x9a97750	107.846	105.309
10000000000ns	0x9a97878	101.312	94.9267
10000000000ns	0x9a979a0	104.955	77.1553
10000000000ns	0x9a97ac8	86.3279	89.716
10000000000ns	0x9a97bf0	104.459	128.785
10000000000ns	0x9a97d18	110.413	91.4406
10000000000ns	0x9a97e58	125.641	100.813
10000000000ns	0x9a97f98	108.806	88.2807
10000000000ns	0x9a980d8	121.285	91.6483
10000000000ns	0x9a98218	115.941	72.2532
10000000000ns	0x9a98358	129.87	109.497

10000000000ns	0x9a98498	83.1673	101.885
10000000000ns	0x9a985d8	102.796	128.813
10000000000ns	0x9a98718	83.6014	108.969
10000000000ns	0x9a98858	98.0811	107.414
10000000000ns	0x9a98998	75.0921	119.66
10000000000ns	0x9a98ad8	95.4459	95.0423
10000000000ns	0x9a98c18	82.6179	100.472
10000000000ns	0x9a98d58	95.335	104.715
10000000000ns	0x9a98e98	95.4191	106.021
10000000000ns	0x9a98fd8	104.985	99.3439
10000000000ns	0x9a99118	92.5262	72.2015
10000000000ns	0x9a99258	90.8443	106.273
10000000000ns	0x9a99398	86.9724	124.159
12000000000ns	0x9a96da8	114.225	75.9294
12000000000ns	0x9a96f98	93.9238	93.9783
12000000000ns	0x9a97080	83.6099	98.3223
12000000000ns	0x9a97188	89.0592	96.4971
12000000000ns	0x9a972b0	90.0233	107.421
12000000000ns	0x9a973d8	113.917	82.5861
12000000000ns	0x9a97500	111.39	119.316
12000000000ns	0x9a97628	80.2643	94.073
12000000000ns	0x9a97750	109.805	104.91
12000000000ns	0x9a97878	101.828	96.8591
12000000000ns	0x9a979a0	103.096	76.4156
12000000000ns	0x9a97ac8	88.3278	89.6919
12000000000ns	0x9a97bf0	104.933	126.842
12000000000ns	0x9a97d18	109.154	92.9945
12000000000ns	0x9a97e58	124.412	99.2348
12000000000ns	0x9a97f98	107.382	89.685
12000000000ns	0x9a980d8	119.621	90.5375
12000000000ns	0x9a98218	114.484	73.623
12000000000ns	0x9a98358	128.588	107.961
12000000000ns	0x9a98498	82.7191	99.9354
12000000000ns	0x9a985d8	103.115	130.787
12000000000ns	0x9a98718	85.6009	109.012
12000000000ns	0x9a98858	96.2031	108.102
12000000000ns	0x9a98998	76.6112	118.359
12000000000ns	0x9a98ad8	96.4218	93.2966
12000000000ns	0x9a98c18	80.652	100.104
12000000000ns	0x9a98d58	93.3795	105.135
12000000000ns	0x9a98e98	97.4083	105.814
12000000000ns	0x9a98fd8	106.903	98.779
12000000000ns	0x9a99118	90.8832	71.061
12000000000ns	0x9a99258	92.558	107.304
12000000000ns	0x9a99398	87.087	126.156

14000000000ns	0x9a96da8	115.23	74.1999
14000000000ns	0x9a96f98	92.0323	94.6283
14000000000ns	0x9a97080	82.4046	96.7264
14000000000ns	0x9a97188	90.7041	95.3594
14000000000ns	0x9a972b0	89.6586	109.387
14000000000ns	0x9a973d8	114.134	80.598
14000000000ns	0x9a97500	110.161	120.893
14000000000ns	0x9a97628	82.0559	93.1842
14000000000ns	0x9a97750	110.383	102.995
14000000000ns	0x9a97878	103.085	98.4145
14000000000ns	0x9a979a0	103.962	74.6128
14000000000ns	0x9a97ac8	88.5605	87.7055
14000000000ns	0x9a97bf0	106.786	127.596
14000000000ns	0x9a97d18	110.909	93.9533
14000000000ns	0x9a97e58	125.901	97.8998
14000000000ns	0x9a97f98	105.64	90.6674
14000000000ns	0x9a980d8	121.608	90.3071
14000000000ns	0x9a98218	114.934	71.6743
14000000000ns	0x9a98358	129.994	109.384
14000000000ns	0x9a98498	81.0197	100.99
14000000000ns	0x9a985d8	101.526	132.002
14000000000ns	0x9a98718	86.5433	110.776
14000000000ns	0x9a98858	97.9019	107.047
14000000000ns	0x9a98998	75.8677	120.216
14000000000ns	0x9a98ad8	98.3813	93.6969
14000000000ns	0x9a98c18	78.7192	99.5904
14000000000ns	0x9a98d58	93.4954	103.138
14000000000ns	0x9a98e98	99.3992	105.623
14000000000ns	0x9a98fd8	105.102	97.9096
14000000000ns	0x9a99118	92.5843	70.0092
14000000000ns	0x9a99258	92.6293	109.302
14000000000ns	0x9a99398	89.0732	126.39
16000000000ns	0x9a96da8	117.101	73.4945
16000000000ns	0x9a96f98	90.4858	95.8965
16000000000ns	0x9a97080	81.4413	98.4791
16000000000ns	0x9a97188	88.801	95.9744
16000000000ns	0x9a972b0	87.6591	109.431
16000000000ns	0x9a973d8	115.539	79.174
16000000000ns	0x9a97500	111.17	122.62
16000000000ns	0x9a97628	83.5204	94.5463
16000000000ns	0x9a97750	108.515	102.281
16000000000ns	0x9a97878	104.218	96.7664
16000000000ns	0x9a979a0	103.358	76.5194
16000000000ns	0x9a97ac8	89.8642	89.2222
16000000000ns	0x9a97bf0	108.643	128.337

16000000000ns	0x9a97d18	109.652	95.5085
16000000000ns	0x9a97e58	127.849	98.3514
16000000000ns	0x9a97f98	104.898	92.5244
16000000000ns	0x9a980d8	121.02	92.2187
16000000000ns	0x9a98218	115.551	73.5767
16000000000ns	0x9a98358	129.148	111.196
16000000000ns	0x9a98498	83.0193	101.034
16000000000ns	0x9a985d8	102.98	130.628
16000000000ns	0x9a98718	86.5776	112.776
16000000000ns	0x9a98858	98.401	105.11
16000000000ns	0x9a98998	74.6128	118.658
16000000000ns	0x9a98ad8	96.3835	93.7914
16000000000ns	0x9a98c18	80.0972	98.1409
16000000000ns	0x9a98d58	92.4088	104.817
16000000000ns	0x9a98e98	98.5941	107.454
16000000000ns	0x9a98fd8	106.985	97.2358
16000000000ns	0x9a99118	93.5046	68.2335
16000000000ns	0x9a99258	93.3838	111.155
16000000000ns	0x9a99398	90.1706	128.062
18000000000ns	0x9a96da8	119.101	73.4959
18000000000ns	0x9a96f98	89.2127	97.4389
18000000000ns	0x9a97080	79.8464	97.2724
18000000000ns	0x9a97188	88.0189	97.8152
18000000000ns	0x9a972b0	88.7037	107.726
18000000000ns	0x9a973d8	113.552	79.4064
18000000000ns	0x9a97500	109.171	122.686
18000000000ns	0x9a97628	84.4978	92.8014
18000000000ns	0x9a97750	108.573	104.28
18000000000ns	0x9a97878	102.589	95.606
18000000000ns	0x9a979a0	103.17	78.5105
18000000000ns	0x9a97ac8	89.204	91.1101
18000000000ns	0x9a97bf0	109.246	126.43
18000000000ns	0x9a97d18	107.841	94.6591
18000000000ns	0x9a97e58	128.004	96.3573
18000000000ns	0x9a97f98	104.968	94.5231
18000000000ns	0x9a980d8	121.446	90.2647
18000000000ns	0x9a98218	115.224	75.5498
18000000000ns	0x9a98358	131.06	110.609
18000000000ns	0x9a98498	84.9334	101.614
18000000000ns	0x9a985d8	104.911	131.15
18000000000ns	0x9a98718	84.6049	112.447
18000000000ns	0x9a98858	98.3001	107.107
18000000000ns	0x9a98998	74.9323	116.684
18000000000ns	0x9a98ad8	97.1054	95.6566
18000000000ns	0x9a98c18	81.0198	99.9154

18000000000ns	0x9a98d58	93.6746	103.269
18000000000ns	0x9a98e98	97.6404	105.696
18000000000ns	0x9a98fd8	108.957	96.9005
18000000000ns	0x9a99118	95.2112	69.2764
18000000000ns	0x9a99258	93.5901	109.165
18000000000ns	0x9a99398	90.3659	126.072
20000000000ns	0x9a96da8	118.42	75.3765
20000000000ns	0x9a96f98	87.5511	98.5521
20000000000ns	0x9a97080	77.9197	97.8091
20000000000ns	0x9a97188	89.155	99.4612
20000000000ns	0x9a972b0	90.6205	108.296
20000000000ns	0x9a973d8	111.86	78.3396
20000000000ns	0x9a97500	108.145	124.403
20000000000ns	0x9a97628	83.1817	91.2954
20000000000ns	0x9a97750	108.365	106.269
20000000000ns	0x9a97878	100.631	96.0133
20000000000ns	0x9a979a0	103.629	76.5638
20000000000ns	0x9a97ac8	87.2049	91.1689
20000000000ns	0x9a97bf0	107.248	126.334
20000000000ns	0x9a97d18	109.412	93.4215
20000000000ns	0x9a97e58	128.057	94.358
20000000000ns	0x9a97f98	105.823	92.7148
20000000000ns	0x9a980d8	119.627	91.0951
20000000000ns	0x9a98218	116.826	76.7483
20000000000ns	0x9a98358	129.934	108.957
20000000000ns	0x9a98498	83.0492	100.943
20000000000ns	0x9a985d8	103.313	132.353
20000000000ns	0x9a98718	82.6251	112.164
20000000000ns	0x9a98858	96.6962	105.912
20000000000ns	0x9a98998	75.1326	118.674
20000000000ns	0x9a98ad8	95.1573	95.204
20000000000ns	0x9a98c18	79.0214	99.8352
20000000000ns	0x9a98d58	95.2732	104.471
20000000000ns	0x9a98e98	99.6342	105.853
20000000000ns	0x9a98fd8	107.03	97.4362
20000000000ns	0x9a99118	96.3727	67.6483
20000000000ns	0x9a99258	95.5023	109.751
20000000000ns	0x9a99398	88.372	125.915
22000000000ns	0x9a96da8	116.474	74.9173
22000000000ns	0x9a96f98	86.0185	99.837
22000000000ns	0x9a97080	76.7361	96.1969
22000000000ns	0x9a97188	88.8886	101.443
22000000000ns	0x9a972b0	90.943	106.323
22000000000ns	0x9a973d8	112.557	76.4648
22000000000ns	0x9a97500	109.127	122.66

22000000000ns	0x9a97628	82.2633	93.072
22000000000ns	0x9a97750	106.643	107.286
22000000000ns	0x9a97878	98.8131	96.8469
22000000000ns	0x9a979a0	102.519	78.2274
22000000000ns	0x9a97ac8	85.7524	89.7941
22000000000ns	0x9a97bf0	105.449	125.46
22000000000ns	0x9a97d18	110.326	91.6427
22000000000ns	0x9a97e58	129.828	93.4292
22000000000ns	0x9a97f98	105.557	94.6971
22000000000ns	0x9a980d8	120.823	89.4921
22000000000ns	0x9a98218	118.476	77.8781
22000000000ns	0x9a98358	131.237	107.44
22000000000ns	0x9a98498	83.8791	99.1235
22000000000ns	0x9a985d8	101.903	133.772
22000000000ns	0x9a98718	83.3345	114.034
22000000000ns	0x9a98858	97.3522	104.023
22000000000ns	0x9a98998	73.1367	118.545
22000000000ns	0x9a98ad8	93.3619	94.3228
22000000000ns	0x9a98c18	77.2457	98.915
22000000000ns	0x9a98d58	93.6109	105.583
22000000000ns	0x9a98e98	98.4247	104.26
22000000000ns	0x9a98fd8	105.913	95.7772
22000000000ns	0x9a99118	96.8528	69.5899
22000000000ns	0x9a99258	93.5467	110.17
22000000000ns	0x9a99398	86.6805	124.848
24000000000ns	0x9a96da8	114.855	76.0922
24000000000ns	0x9a96f98	84.5262	101.169
24000000000ns	0x9a97080	76.0535	98.0768
24000000000ns	0x9a97188	89.1231	103.43
24000000000ns	0x9a972b0	92.9355	106.496
24000000000ns	0x9a973d8	112.667	78.4618
24000000000ns	0x9a97500	110.6	124.013
24000000000ns	0x9a97628	82.2556	91.072
24000000000ns	0x9a97750	108.036	108.722
24000000000ns	0x9a97878	98.5982	94.8585
24000000000ns	0x9a979a0	101.218	79.7464
24000000000ns	0x9a97ac8	87.6132	89.0611
24000000000ns	0x9a97bf0	104.323	127.112
24000000000ns	0x9a97d18	112.077	90.6752
24000000000ns	0x9a97e58	130.716	91.637
24000000000ns	0x9a97f98	107.325	95.632
24000000000ns	0x9a980d8	120.257	87.574
24000000000ns	0x9a98218	118.883	79.8362
24000000000ns	0x9a98358	130.519	109.306
24000000000ns	0x9a98498	85.8787	99.1656

24000000000ns	0x9a985d8	99.9602	133.298
24000000000ns	0x9a98718	85.3082	113.711
24000000000ns	0x9a98858	96.9381	105.98
24000000000ns	0x9a98998	72.7733	120.511
24000000000ns	0x9a98ad8	95.1852	95.1447
24000000000ns	0x9a98c18	78.6598	100.329
24000000000ns	0x9a98d58	93.7182	103.586
24000000000ns	0x9a98e98	99.1684	102.404
24000000000ns	0x9a98fd8	107.181	97.3241
24000000000ns	0x9a99118	96.3425	71.5237
24000000000ns	0x9a99258	92.2981	111.733
24000000000ns	0x9a99398	88.1753	126.177
26000000000ns	0x9a96da8	116.537	77.1747
26000000000ns	0x9a96f98	86.4129	100.505
26000000000ns	0x9a97080	74.6298	96.6722
26000000000ns	0x9a97188	91.1193	103.305
26000000000ns	0x9a972b0	92.3698	108.414
26000000000ns	0x9a973d8	113.574	80.2443
26000000000ns	0x9a97500	109.287	125.522
26000000000ns	0x9a97628	83.205	92.8323
26000000000ns	0x9a97750	106.782	107.163
26000000000ns	0x9a97878	97.4654	93.2102
26000000000ns	0x9a979a0	100.746	77.803
26000000000ns	0x9a97ac8	89.6127	89.0141
26000000000ns	0x9a97bf0	106.142	127.944
26000000000ns	0x9a97d18	110.228	89.9112
26000000000ns	0x9a97e58	128.721	91.7832
26000000000ns	0x9a97f98	105.493	96.4361
26000000000ns	0x9a980d8	118.568	88.6462
26000000000ns	0x9a98218	117.051	79.0347
26000000000ns	0x9a98358	130.046	111.25
26000000000ns	0x9a98498	87.3875	100.478
26000000000ns	0x9a985d8	99.3717	135.209
26000000000ns	0x9a98718	85.7644	111.764
26000000000ns	0x9a98858	95.3064	107.136
26000000000ns	0x9a98998	70.8074	120.879
26000000000ns	0x9a98ad8	93.3446	95.927
26000000000ns	0x9a98c18	76.7657	99.6872
26000000000ns	0x9a98d58	94.9856	102.038
26000000000ns	0x9a98e98	97.1956	102.075
26000000000ns	0x9a98fd8	105.543	98.4725
26000000000ns	0x9a99118	95.9041	73.475
26000000000ns	0x9a99258	94.2452	112.189
26000000000ns	0x9a99398	90.0916	126.75
28000000000ns	0x9a96da8	115.342	75.5713

28000000000ns	0x9a96f98	85.9407	102.448
28000000000ns	0x9a97080	73.6829	98.4339
28000000000ns	0x9a97188	92.011	101.515
28000000000ns	0x9a972b0	93.5651	106.811
28000000000ns	0x9a973d8	115.368	81.1291
28000000000ns	0x9a97500	111.287	125.53
28000000000ns	0x9a97628	82.2419	91.0795
28000000000ns	0x9a97750	107.823	108.871
28000000000ns	0x9a97878	97.4658	91.2102
28000000000ns	0x9a979a0	98.7494	77.9238
28000000000ns	0x9a97ac8	91.6075	89.1581
28000000000ns	0x9a97bf0	104.143	127.883
28000000000ns	0x9a97d18	110.47	91.8966
28000000000ns	0x9a97e58	126.768	92.2144
28000000000ns	0x9a97f98	103.504	96.229
28000000000ns	0x9a980d8	116.57	88.5551
28000000000ns	0x9a98218	118.012	77.2808
28000000000ns	0x9a98358	132.028	111.52
28000000000ns	0x9a98498	87.0663	98.5044
28000000000ns	0x9a985d8	100.834	136.574
28000000000ns	0x9a98718	84.6773	113.442
28000000000ns	0x9a98858	96.3664	108.832
28000000000ns	0x9a98998	68.9399	121.595
28000000000ns	0x9a98ad8	95.2495	95.3178
28000000000ns	0x9a98c18	75.6235	98.0455
28000000000ns	0x9a98d58	95.4457	100.092
28000000000ns	0x9a98e98	95.7088	100.737
28000000000ns	0x9a98fd8	103.719	97.6535
28000000000ns	0x9a99118	94.1693	74.4702
28000000000ns	0x9a99258	95.3454	113.86
28000000000ns	0x9a99398	89.8083	124.77
30000000000ns	0x9a96da8	115.826	77.5117
30000000000ns	0x9a96f98	87.4868	101.18
30000000000ns	0x9a97080	72.9275	100.286
30000000000ns	0x9a97188	92.803	103.352
30000000000ns	0x9a972b0	92.6743	105.02
30000000000ns	0x9a973d8	115.746	83.093
30000000000ns	0x9a97500	110.672	123.627
30000000000ns	0x9a97628	83.3696	92.7312
30000000000ns	0x9a97750	106.249	110.105
30000000000ns	0x9a97878	96.0474	89.8001
30000000000ns	0x9a979a0	100.401	76.7955
30000000000ns	0x9a97ac8	90.67	90.9248
30000000000ns	0x9a97bf0	104.913	126.037
30000000000ns	0x9a97d18	112.352	92.5741

30000000000ns	0x9a97e58	125.418	90.7387
30000000000ns	0x9a97f98	102.098	97.6516
30000000000ns	0x9a980d8	116.763	90.5458
30000000000ns	0x9a98218	119.83	76.4465
30000000000ns	0x9a98358	130.281	110.547
30000000000ns	0x9a98498	89.0663	98.5005
30000000000ns	0x9a985d8	99.1646	135.472
30000000000ns	0x9a98718	84.6984	111.442
30000000000ns	0x9a98858	98.2488	109.508
30000000000ns	0x9a98998	70.3113	120.139
30000000000ns	0x9a98ad8	96.6339	93.8743
30000000000ns	0x9a98c18	73.8297	98.9299
30000000000ns	0x9a98d58	96.0106	98.1735
30000000000ns	0x9a98e98	95.6485	102.736
30000000000ns	0x9a98fd8	104.796	99.3385
30000000000ns	0x9a99118	94.7734	76.3768
30000000000ns	0x9a99258	94.7729	115.776
30000000000ns	0x9a99398	90.6063	122.936
32000000000ns	0x9a96da8	115.048	75.6692
32000000000ns	0x9a96f98	86.01	99.8309
32000000000ns	0x9a97080	73.8672	102.051
32000000000ns	0x9a97188	94.1985	104.784
32000000000ns	0x9a972b0	94.309	103.868
32000000000ns	0x9a973d8	117.618	82.3887
32000000000ns	0x9a97500	109.985	121.749
32000000000ns	0x9a97628	81.5556	93.5734
32000000000ns	0x9a97750	104.972	111.644
32000000000ns	0x9a97878	94.549	91.1247
32000000000ns	0x9a979a0	101.203	78.6277
32000000000ns	0x9a97ac8	92.1627	89.5936
32000000000ns	0x9a97bf0	105.141	128.024
32000000000ns	0x9a97d18	112.67	90.5996
32000000000ns	0x9a97e58	125.725	88.7624
32000000000ns	0x9a97f98	100.221	98.3413
32000000000ns	0x9a980d8	114.766	90.6514
32000000000ns	0x9a98218	117.84	76.6496
32000000000ns	0x9a98358	129.571	112.417
32000000000ns	0x9a98498	87.682	97.057
32000000000ns	0x9a985d8	97.4735	136.54
32000000000ns	0x9a98718	86.2396	110.168
32000000000ns	0x9a98858	100.06	108.66
32000000000ns	0x9a98998	69.0869	121.72
32000000000ns	0x9a98ad8	98.5058	94.5786
32000000000ns	0x9a98c18	74.8582	100.645
32000000000ns	0x9a98d58	97.3879	99.6236

32000000000ns	0x9a98e98	93.7771	102.031
32000000000ns	0x9a98fd8	102.796	99.328
32000000000ns	0x9a99118	93.2154	75.1228
32000000000ns	0x9a99258	94.836	117.775
32000000000ns	0x9a99398	90.2827	124.909
34000000000ns	0x9a96da8	116.932	74.9967
34000000000ns	0x9a96f98	85.0489	98.077
34000000000ns	0x9a97080	72.8329	100.339
34000000000ns	0x9a97188	92.4983	103.731
34000000000ns	0x9a972b0	96.2645	104.287
34000000000ns	0x9a973d8	119.507	83.0463
34000000000ns	0x9a97500	109.163	123.572
34000000000ns	0x9a97628	83.5368	93.3001
34000000000ns	0x9a97750	106.899	112.18
34000000000ns	0x9a97878	94.085	93.0702
34000000000ns	0x9a979a0	102.788	79.8464
34000000000ns	0x9a97ac8	91.9366	87.6065
34000000000ns	0x9a97bf0	103.805	129.513
34000000000ns	0x9a97d18	114.579	91.1966
34000000000ns	0x9a97e58	124.467	90.3169
34000000000ns	0x9a97f98	99.7772	100.291
34000000000ns	0x9a980d8	113.121	91.7892
34000000000ns	0x9a98218	119.478	75.5017
34000000000ns	0x9a98358	129.753	114.409
34000000000ns	0x9a98498	89.3652	98.1372
34000000000ns	0x9a985d8	99.1213	135.406
34000000000ns	0x9a98718	85.1263	111.829
34000000000ns	0x9a98858	100.398	110.631
34000000000ns	0x9a98998	67.3196	122.657
34000000000ns	0x9a98ad8	100.071	93.3341
34000000000ns	0x9a98c18	76.281	102.051
34000000000ns	0x9a98d58	96.615	101.468
34000000000ns	0x9a98e98	93.2512	103.96
34000000000ns	0x9a98fd8	104.044	97.7647
34000000000ns	0x9a99118	93.7624	73.1991
34000000000ns	0x9a99258	95.2493	115.818
34000000000ns	0x9a99398	89.1006	126.523
36000000000ns	0x9a96da8	117.621	76.874
36000000000ns	0x9a96f98	86.5358	99.4146
36000000000ns	0x9a97080	73.2656	98.3868
36000000000ns	0x9a97188	94.4509	104.164
36000000000ns	0x9a972b0	97.8593	103.08
36000000000ns	0x9a973d8	121.431	82.4995
36000000000ns	0x9a97500	107.177	123.33
36000000000ns	0x9a97628	85.3117	92.3782

36000000000ns	0x9a97750	106.999	114.177
36000000000ns	0x9a97878	95.8803	93.9516
36000000000ns	0x9a979a0	101.099	78.776
36000000000ns	0x9a97ac8	90.451	88.9455
36000000000ns	0x9a97bf0	101.844	129.907
36000000000ns	0x9a97d18	116.225	90.061
36000000000ns	0x9a97e58	122.477	90.1115
36000000000ns	0x9a97f98	101.773	100.167
36000000000ns	0x9a980d8	114.877	90.8324
36000000000ns	0x9a98218	117.478	75.4788
36000000000ns	0x9a98358	128.68	116.097
36000000000ns	0x9a98498	90.1349	96.2912
36000000000ns	0x9a985d8	97.1311	135.21
36000000000ns	0x9a98718	86.6825	110.573
36000000000ns	0x9a98858	101.07	112.515
36000000000ns	0x9a98998	66.1471	124.277
36000000000ns	0x9a98ad8	98.852	94.9193
36000000000ns	0x9a98c18	74.6283	103.177
36000000000ns	0x9a98d58	98.4127	102.345
36000000000ns	0x9a98e98	91.3319	103.398
36000000000ns	0x9a98fd8	105.407	96.3012
36000000000ns	0x9a99118	91.7627	73.2316
36000000000ns	0x9a99258	94.6349	117.721
36000000000ns	0x9a99398	87.6158	125.183
38000000000ns	0x9a96da8	119.477	76.1277
38000000000ns	0x9a96f98	86.5231	97.4146
38000000000ns	0x9a97080	75.2516	98.1502
38000000000ns	0x9a97188	92.6802	103.234
38000000000ns	0x9a972b0	99.8593	103.077
38000000000ns	0x9a973d8	120.629	80.667
38000000000ns	0x9a97500	107.024	121.336
38000000000ns	0x9a97628	84.396	90.6001
38000000000ns	0x9a97750	108.979	113.897
38000000000ns	0x9a97878	94.0771	93.0865
38000000000ns	0x9a979a0	101.866	76.929
38000000000ns	0x9a97ac8	92.0857	90.0979
38000000000ns	0x9a97bf0	101.68	131.9
38000000000ns	0x9a97d18	114.225	90.0612
38000000000ns	0x9a97e58	120.602	90.8048
38000000000ns	0x9a97f98	103.688	99.5908
38000000000ns	0x9a980d8	116.4	92.1289
38000000000ns	0x9a98218	115.596	74.8028
38000000000ns	0x9a98358	130.647	115.734
38000000000ns	0x9a98498	90.5153	98.2547
38000000000ns	0x9a985d8	99.0944	134.828

38000000000ns	0x9a98718	88.5361	109.822
38000000000ns	0x9a98858	99.0709	112.454
38000000000ns	0x9a98998	66.2138	126.276
38000000000ns	0x9a98ad8	100.654	94.0515
38000000000ns	0x9a98c18	76.5012	103.879
38000000000ns	0x9a98d58	99.7626	103.82
38000000000ns	0x9a98e98	91.3969	101.399
38000000000ns	0x9a98fd8	103.407	96.2895
38000000000ns	0x9a99118	90.8491	75.0108
38000000000ns	0x9a99258	96.6105	118.033
38000000000ns	0x9a99398	89.5763	125.578
40000000000ns	0x9a96da8	117.763	75.0963
40000000000ns	0x9a96f98	87.1877	95.5283
40000000000ns	0x9a97080	76.0438	96.3138
40000000000ns	0x9a97188	94.3323	104.361
40000000000ns	0x9a972b0	101.553	102.014
40000000000ns	0x9a973d8	122.078	82.046
40000000000ns	0x9a97500	106.597	119.383
40000000000ns	0x9a97628	85.3417	88.8378
40000000000ns	0x9a97750	107.277	114.946
40000000000ns	0x9a97878	94.5227	95.0362
40000000000ns	0x9a979a0	103.769	77.544
40000000000ns	0x9a97ac8	92.5054	92.0533
40000000000ns	0x9a97bf0	103.222	130.626
40000000000ns	0x9a97d18	112.65	88.8287
40000000000ns	0x9a97e58	122.596	90.9574
40000000000ns	0x9a97f98	105.542	100.341
40000000000ns	0x9a980d8	117.175	90.2851
40000000000ns	0x9a98218	117.578	75.0717
40000000000ns	0x9a98358	131.221	113.818
40000000000ns	0x9a98498	91.175	96.3666
40000000000ns	0x9a985d8	100.755	135.942
40000000000ns	0x9a98718	87.4629	108.134
40000000000ns	0x9a98858	99.997	114.226
40000000000ns	0x9a98998	67.5449	127.769
40000000000ns	0x9a98ad8	102.198	95.3231
40000000000ns	0x9a98c18	74.8635	102.731
40000000000ns	0x9a98d58	99.6778	105.819
40000000000ns	0x9a98e98	89.4022	101.545
40000000000ns	0x9a98fd8	103.27	94.2942
40000000000ns	0x9a99118	90.4743	73.0462
40000000000ns	0x9a99258	95.367	119.599
40000000000ns	0x9a99398	90.8151	127.149
42000000000ns	0x9a96da8	118.847	73.4153
42000000000ns	0x9a96f98	85.1982	95.3233

42000000000ns	0x9a97080	74.5016	95.0404
42000000000ns	0x9a97188	95.2615	106.132
42000000000ns	0x9a972b0	102.303	100.16
42000000000ns	0x9a973d8	120.167	82.6363
42000000000ns	0x9a97500	108.259	118.271
42000000000ns	0x9a97628	85.3224	90.8377
42000000000ns	0x9a97750	108.731	116.319
42000000000ns	0x9a97878	94.123	96.9959
42000000000ns	0x9a979a0	102.857	79.3241
42000000000ns	0x9a97ac8	90.5353	91.7088
42000000000ns	0x9a97bf0	101.527	131.688
42000000000ns	0x9a97d18	110.716	88.3193
42000000000ns	0x9a97e58	122.317	88.9769
42000000000ns	0x9a97f98	103.751	99.4515
42000000000ns	0x9a980d8	118.806	89.1273
42000000000ns	0x9a98218	119.068	76.4055
42000000000ns	0x9a98358	131.594	111.854
42000000000ns	0x9a98498	92.1608	98.1067
42000000000ns	0x9a985d8	98.8582	136.575
42000000000ns	0x9a98718	86.447	109.857
42000000000ns	0x9a98858	100.668	112.342
42000000000ns	0x9a98998	65.7785	128.707
42000000000ns	0x9a98ad8	103.283	97.0032
42000000000ns	0x9a98c18	73.911	100.972
42000000000ns	0x9a98d58	100.357	103.937
42000000000ns	0x9a98e98	87.4286	101.221
42000000000ns	0x9a98fd8	103.894	92.3943
42000000000ns	0x9a99118	88.7096	72.105
42000000000ns	0x9a99258	93.3878	119.887
42000000000ns	0x9a99398	89.202	125.966
44000000000ns	0x9a96da8	118.42	75.3692
44000000000ns	0x9a96f98	87.0316	94.524
44000000000ns	0x9a97080	76.33	95.8509
44000000000ns	0x9a97188	93.4243	106.923
44000000000ns	0x9a972b0	102.236	102.159
44000000000ns	0x9a973d8	118.951	81.048
44000000000ns	0x9a97500	108.269	120.271
44000000000ns	0x9a97628	85.1802	92.8326
44000000000ns	0x9a97750	108.793	114.32
44000000000ns	0x9a97878	93.4709	95.1052
44000000000ns	0x9a979a0	104.757	78.6988
44000000000ns	0x9a97ac8	90.3547	89.7169
44000000000ns	0x9a97bf0	99.5903	131.19
44000000000ns	0x9a97d18	109.365	89.7943
44000000000ns	0x9a97e58	123.78	90.3406

44000000000ns	0x9a97f98	105.623	98.747
44000000000ns	0x9a980d8	119.24	91.0797
44000000000ns	0x9a98218	120.906	77.1937
44000000000ns	0x9a98358	132.45	113.661
44000000000ns	0x9a98498	90.8609	96.5868
44000000000ns	0x9a985d8	100.628	137.506
44000000000ns	0x9a98718	88.4425	109.722
44000000000ns	0x9a98858	99.1055	113.59
44000000000ns	0x9a98998	63.7881	128.511
44000000000ns	0x9a98ad8	101.855	95.6025
44000000000ns	0x9a98c18	73.2986	99.0683
44000000000ns	0x9a98d58	98.4681	104.595
44000000000ns	0x9a98e98	89.3884	101.62
44000000000ns	0x9a98fd8	105.875	92.6689
44000000000ns	0x9a99118	86.8695	71.3213
44000000000ns	0x9a99258	93.1657	117.899
44000000000ns	0x9a99398	88.4626	124.108
46000000000ns	0x9a96da8	119.043	73.4686
46000000000ns	0x9a96f98	88.2287	92.9219
46000000000ns	0x9a97080	77.2107	97.6466
46000000000ns	0x9a97188	94.5729	105.286
46000000000ns	0x9a972b0	102.327	100.161
46000000000ns	0x9a973d8	120.181	82.6254
46000000000ns	0x9a97500	107.96	122.247
46000000000ns	0x9a97628	83.4861	91.7696
46000000000ns	0x9a97750	109.572	116.162
46000000000ns	0x9a97878	95.4689	95.0151
46000000000ns	0x9a979a0	105.669	80.4788
46000000000ns	0x9a97ac8	89.6845	91.6013
46000000000ns	0x9a97bf0	100.449	129.384
46000000000ns	0x9a97d18	108.577	87.9564
46000000000ns	0x9a97e58	121.865	90.9182
46000000000ns	0x9a97f98	107.531	99.3466
46000000000ns	0x9a980d8	121.237	91.1829
46000000000ns	0x9a98218	122.863	76.78
46000000000ns	0x9a98358	131.963	115.601
46000000000ns	0x9a98498	91.5244	94.7001
46000000000ns	0x9a985d8	99.869	139.356
46000000000ns	0x9a98718	89.7267	111.256
46000000000ns	0x9a98858	100.791	114.667
46000000000ns	0x9a98998	65.6236	129.305
46000000000ns	0x9a98ad8	102.63	93.7587
46000000000ns	0x9a98c18	72.1442	100.701
46000000000ns	0x9a98d58	100.35	105.27
46000000000ns	0x9a98e98	90.2376	99.8091

46000000000ns	0x9a98fd8	103.889	92.4319
46000000000ns	0x9a99118	87.1382	69.3394
46000000000ns	0x9a99258	92.9147	119.884
46000000000ns	0x9a99398	88.3531	126.105
48000000000ns	0x9a96da8	121.031	73.687
48000000000ns	0x9a96f98	88.2273	94.9219
48000000000ns	0x9a97080	77.7669	99.5677
48000000000ns	0x9a97188	93.7504	103.462
48000000000ns	0x9a972b0	103.888	98.9107
48000000000ns	0x9a973d8	121.184	84.3557
48000000000ns	0x9a97500	109.749	121.353
48000000000ns	0x9a97628	83.8808	89.8089
48000000000ns	0x9a97750	109.836	118.144
48000000000ns	0x9a97878	97.0625	96.2235
48000000000ns	0x9a979a0	103.739	81.0041
48000000000ns	0x9a97ac8	91.5575	90.9001
48000000000ns	0x9a97bf0	100.406	131.384
48000000000ns	0x9a97d18	109.711	86.3093
48000000000ns	0x9a97e58	123.549	91.9967
48000000000ns	0x9a97f98	109.529	99.4306
48000000000ns	0x9a980d8	120.829	93.1409
48000000000ns	0x9a98218	121.058	75.9184
48000000000ns	0x9a98358	133.298	114.111
48000000000ns	0x9a98498	93.4112	95.3634
48000000000ns	0x9a985d8	98.5955	137.814
48000000000ns	0x9a98718	88.3689	109.787
48000000000ns	0x9a98858	100.126	116.553
48000000000ns	0x9a98998	65.9306	127.329
48000000000ns	0x9a98ad8	100.638	93.9372
48000000000ns	0x9a98c18	73.2117	99.0102
48000000000ns	0x9a98d58	101.714	106.734
48000000000ns	0x9a98e98	88.2574	99.528
48000000000ns	0x9a98fd8	101.896	92.268
48000000000ns	0x9a99118	89.133	69.1956
48000000000ns	0x9a99258	90.9147	119.878
48000000000ns	0x9a99398	87.9425	128.062
50000000000ns	0x9a96da8	121.016	71.6871
50000000000ns	0x9a96f98	89.5676	93.4374
50000000000ns	0x9a97080	76.2047	100.816
50000000000ns	0x9a97188	95.6484	102.832
50000000000ns	0x9a972b0	105.116	97.3324
50000000000ns	0x9a973d8	119.715	82.9979
50000000000ns	0x9a97500	108.511	122.924
50000000000ns	0x9a97628	82.1388	90.7914
50000000000ns	0x9a97750	108.906	119.915

50000000000ns	0x9a97878	95.2787	95.3191
50000000000ns	0x9a979a0	104.089	82.9731
50000000000ns	0x9a97ac8	93.1701	92.0832
50000000000ns	0x9a97bf0	98.6055	130.513
50000000000ns	0x9a97d18	109.168	84.3845
50000000000ns	0x9a97e58	121.641	91.3985
50000000000ns	0x9a97f98	111.506	99.1286
50000000000ns	0x9a980d8	122.199	91.6836
50000000000ns	0x9a98218	121.5	77.8689
50000000000ns	0x9a98358	132.979	112.137
50000000000ns	0x9a98498	94.4485	97.0734
50000000000ns	0x9a985d8	97.5179	136.129
50000000000ns	0x9a98718	90.1706	108.919
50000000000ns	0x9a98858	99.663	114.607
50000000000ns	0x9a98998	64.9548	129.074
50000000000ns	0x9a98ad8	100.549	95.9353
50000000000ns	0x9a98c18	72.9504	100.993
50000000000ns	0x9a98d58	103.555	107.514
50000000000ns	0x9a98e98	87.838	101.484
50000000000ns	0x9a98fd8	99.9149	91.9945
50000000000ns	0x9a99118	89.2782	71.1903
50000000000ns	0x9a99258	92.8997	119.634
50000000000ns	0x9a99398	87.9284	126.062
52000000000ns	0x9a96da8	122.69	70.5924
52000000000ns	0x9a96f98	90.5724	91.7082
52000000000ns	0x9a97080	77.0681	102.62
52000000000ns	0x9a97188	93.6599	103.046
52000000000ns	0x9a972b0	105.757	99.227
52000000000ns	0x9a973d8	118.548	81.3742
52000000000ns	0x9a97500	109.968	121.554
52000000000ns	0x9a97628	82.7304	88.881
52000000000ns	0x9a97750	110.596	120.984
52000000000ns	0x9a97878	94.9183	93.3519
52000000000ns	0x9a979a0	102.988	84.6423
52000000000ns	0x9a97ac8	94.6847	93.3892
52000000000ns	0x9a97bf0	97.1692	131.904
52000000000ns	0x9a97d18	110.663	85.7133
52000000000ns	0x9a97e58	122.516	89.6002
52000000000ns	0x9a97f98	109.638	99.8422
52000000000ns	0x9a980d8	122.635	93.6356
52000000000ns	0x9a98218	123.165	76.7598
52000000000ns	0x9a98358	133.924	110.374
52000000000ns	0x9a98498	94.9619	95.1404
52000000000ns	0x9a985d8	97.0536	134.184
52000000000ns	0x9a98718	92.1693	108.993

52000000000ns	0x9a98858	100.359	112.732
52000000000ns	0x9a98998	65.5835	127.176
52000000000ns	0x9a98ad8	99.1998	94.4593
52000000000ns	0x9a98c18	71.6208	99.499
52000000000ns	0x9a98d58	103.197	109.482
52000000000ns	0x9a98e98	87.4213	103.44
52000000000ns	0x9a98fd8	98.15	91.0538
52000000000ns	0x9a99118	90.0392	69.3408
52000000000ns	0x9a99258	92.2727	117.735
52000000000ns	0x9a99398	88.7372	124.233
54000000000ns	0x9a96da8	120.692	70.515
54000000000ns	0x9a96f98	92.5685	91.8326
54000000000ns	0x9a97080	76.5334	100.693
54000000000ns	0x9a97188	94.3106	104.937
54000000000ns	0x9a972b0	105.677	101.225
54000000000ns	0x9a973d8	117.175	79.9201
54000000000ns	0x9a97500	107.973	121.685
54000000000ns	0x9a97628	84.4427	89.9146
54000000000ns	0x9a97750	108.612	120.731
54000000000ns	0x9a97878	96.7319	94.1951
54000000000ns	0x9a979a0	102.302	82.7635
54000000000ns	0x9a97ac8	96.3831	94.4454
54000000000ns	0x9a97bf0	98.622	133.279
54000000000ns	0x9a97d18	109.59	84.0253
54000000000ns	0x9a97e58	121.483	87.8878
54000000000ns	0x9a97f98	111.572	99.3329
54000000000ns	0x9a980d8	121.026	92.4479
54000000000ns	0x9a98218	123.428	74.7772
54000000000ns	0x9a98358	135.771	111.141
54000000000ns	0x9a98498	95.6875	93.2766
54000000000ns	0x9a985d8	97.5689	136.116
54000000000ns	0x9a98718	93.0187	107.182
54000000000ns	0x9a98858	101.004	114.626
54000000000ns	0x9a98998	66.106	129.106
54000000000ns	0x9a98ad8	99.4529	96.4432
54000000000ns	0x9a98c18	73.1419	98.2004
54000000000ns	0x9a98d58	101.871	107.985
54000000000ns	0x9a98e98	86.4598	105.193
54000000000ns	0x9a98fd8	98.8068	89.1647
54000000000ns	0x9a99118	89.5308	67.4065
54000000000ns	0x9a99258	94.0675	116.852
54000000000ns	0x9a99398	88.446	126.212
56000000000ns	0x9a96da8	119.694	68.7817
56000000000ns	0x9a96f98	93.5567	93.5715
56000000000ns	0x9a97080	75.7836	98.8391

56000000000ns	0x9a97188	94.0142	106.915
56000000000ns	0x9a972b0	104.027	102.356
56000000000ns	0x9a973d8	117.429	81.9038
56000000000ns	0x9a97500	107.117	119.877
56000000000ns	0x9a97628	85.1875	91.7707
56000000000ns	0x9a97750	107.315	119.209
56000000000ns	0x9a97878	95.7448	92.4557
56000000000ns	0x9a979a0	103.335	84.4762
56000000000ns	0x9a97ac8	94.7206	93.3336
56000000000ns	0x9a97bf0	99.8948	134.822
56000000000ns	0x9a97d18	110.727	85.6707
56000000000ns	0x9a97e58	123.468	87.6437
56000000000ns	0x9a97f98	110.1	97.9784
56000000000ns	0x9a980d8	122.278	94.007
56000000000ns	0x9a98218	123.824	76.7376
56000000000ns	0x9a98358	134.046	112.152
56000000000ns	0x9a98498	97.0426	91.8058
56000000000ns	0x9a985d8	96.0746	134.787
56000000000ns	0x9a98718	94.556	108.461
56000000000ns	0x9a98858	100.828	116.618
56000000000ns	0x9a98998	65.7084	127.146
56000000000ns	0x9a98ad8	101.372	97.0077
56000000000ns	0x9a98c18	74.0032	100.005
56000000000ns	0x9a98d58	99.8709	107.95
56000000000ns	0x9a98e98	88.4465	104.963
56000000000ns	0x9a98fd8	100.703	88.5297
56000000000ns	0x9a99118	89.0946	69.3583
56000000000ns	0x9a99258	94.71	114.958
56000000000ns	0x9a99398	89.6658	124.627
58000000000ns	0x9a96da8	121.693	68.7316
58000000000ns	0x9a96f98	94.969	94.9876
58000000000ns	0x9a97080	74.7629	97.1192
58000000000ns	0x9a97188	95.5779	105.668
58000000000ns	0x9a972b0	102.936	104.032
58000000000ns	0x9a973d8	119.414	82.1495
58000000000ns	0x9a97500	108.439	118.376
58000000000ns	0x9a97628	85.42	89.7842
58000000000ns	0x9a97750	105.349	118.84
58000000000ns	0x9a97878	93.9851	91.5051
58000000000ns	0x9a979a0	102.437	82.6891
58000000000ns	0x9a97ac8	95.7161	95.0683
58000000000ns	0x9a97bf0	101.865	134.478
58000000000ns	0x9a97d18	108.94	86.5697
58000000000ns	0x9a97e58	125.257	88.5387
58000000000ns	0x9a97f98	109.075	99.6958

58000000000ns	0x9a980d8	123.22	95.7714
58000000000ns	0x9a98218	124.485	78.6253
58000000000ns	0x9a98358	132.288	111.198
58000000000ns	0x9a98498	95.8006	90.2382
58000000000ns	0x9a985d8	94.8656	133.194
58000000000ns	0x9a98718	95.5435	110.201
58000000000ns	0x9a98858	100.931	118.615
58000000000ns	0x9a98998	63.8211	126.484
58000000000ns	0x9a98ad8	100.669	98.8804
58000000000ns	0x9a98c18	75.8158	100.851
58000000000ns	0x9a98d58	99.4433	105.997
58000000000ns	0x9a98e98	88.4943	106.962
58000000000ns	0x9a98fd8	99.9074	86.695
58000000000ns	0x9a99118	90.3457	70.9187
58000000000ns	0x9a99258	92.9938	113.931
58000000000ns	0x9a99398	87.6898	124.936
60000000000ns	0x9a96da8	122.427	70.5921
60000000000ns	0x9a96f98	96.8301	95.7199
60000000000ns	0x9a97080	73.7749	98.8581
60000000000ns	0x9a97188	94.07	104.354
60000000000ns	0x9a972b0	100.954	104.301
60000000000ns	0x9a973d8	120.18	83.9972
60000000000ns	0x9a97500	108.204	116.39
60000000000ns	0x9a97628	85.7188	91.7618
60000000000ns	0x9a97750	105.198	116.846
60000000000ns	0x9a97878	93.3286	89.6159
60000000000ns	0x9a979a0	103.348	84.4696
60000000000ns	0x9a97ac8	96.2862	96.9853
60000000000ns	0x9a97bf0	99.8761	134.269
60000000000ns	0x9a97d18	108.643	88.5475
60000000000ns	0x9a97e58	127.256	88.4982
60000000000ns	0x9a97f98	111.056	99.4196
60000000000ns	0x9a980d8	124.774	97.0311
60000000000ns	0x9a98218	125.757	80.1684
60000000000ns	0x9a98358	133.051	109.35
60000000000ns	0x9a98498	95.3917	88.2804
60000000000ns	0x9a985d8	93.9053	134.948
60000000000ns	0x9a98718	95.6074	112.2
60000000000ns	0x9a98858	101.814	120.41
60000000000ns	0x9a98998	62.6114	124.892
60000000000ns	0x9a98ad8	101.387	100.747
60000000000ns	0x9a98c18	76.3489	102.778
60000000000ns	0x9a98d58	101.418	106.311
60000000000ns	0x9a98e98	88.8973	108.921
60000000000ns	0x9a98fd8	101.09	88.3082

60000000000ns	0x9a99118	89.2908	72.6178
60000000000ns	0x9a99258	92.87	111.935
60000000000ns	0x9a99398	86.9479	126.793
62000000000ns	0x9a96da8	123.02	72.5021
62000000000ns	0x9a96f98	98.5143	96.7986
62000000000ns	0x9a97080	74.1636	96.8962
62000000000ns	0x9a97188	92.7117	102.886
62000000000ns	0x9a972b0	102.954	104.277
62000000000ns	0x9a973d8	120.883	82.1247
62000000000ns	0x9a97500	109.155	114.631
62000000000ns	0x9a97628	85.1097	93.6668
62000000000ns	0x9a97750	106.884	117.922
62000000000ns	0x9a97878	92.877	91.5643
62000000000ns	0x9a979a0	103.179	82.4767
62000000000ns	0x9a97ac8	98.2483	97.3727
62000000000ns	0x9a97bf0	101.059	132.656
62000000000ns	0x9a97d18	107.891	90.4007
62000000000ns	0x9a97e58	129.105	89.2607
62000000000ns	0x9a97f98	109.702	97.9479
62000000000ns	0x9a980d8	126.409	98.183
62000000000ns	0x9a98218	123.782	79.8525
62000000000ns	0x9a98358	132.12	111.12
62000000000ns	0x9a98498	97.3503	88.6851
62000000000ns	0x9a985d8	95.549	136.088
62000000000ns	0x9a98718	93.9806	113.363
62000000000ns	0x9a98858	103.721	119.804
62000000000ns	0x9a98998	60.6195	125.071
62000000000ns	0x9a98ad8	100.086	99.2277
62000000000ns	0x9a98c18	77.7562	104.199
62000000000ns	0x9a98d58	102.102	104.432
62000000000ns	0x9a98e98	86.9474	108.476
62000000000ns	0x9a98fd8	99.4965	87.0989
62000000000ns	0x9a99118	87.8045	71.2796
62000000000ns	0x9a99258	92.5093	113.902
62000000000ns	0x9a99398	88.0284	125.11
64000000000ns	0x9a96da8	123.466	74.4519
64000000000ns	0x9a96f98	100.128	97.9806
64000000000ns	0x9a97080	74.3131	98.8906
64000000000ns	0x9a97188	90.7613	103.329
64000000000ns	0x9a972b0	100.954	104.336
64000000000ns	0x9a973d8	122.209	80.6275
64000000000ns	0x9a97500	109.61	112.683
64000000000ns	0x9a97628	83.1796	93.1426
64000000000ns	0x9a97750	108.007	116.268
64000000000ns	0x9a97878	90.8841	91.3957

64000000000ns	0x9a979a0	101.248	82.9961
64000000000ns	0x9a97ac8	96.2541	97.5248
64000000000ns	0x9a97bf0	102.931	133.359
64000000000ns	0x9a97d18	106.189	89.3507
64000000000ns	0x9a97e58	127.894	90.8527
64000000000ns	0x9a97f98	110.372	99.8325
64000000000ns	0x9a980d8	126.297	100.18
64000000000ns	0x9a98218	124.711	78.0816
64000000000ns	0x9a98358	132.6	109.178
64000000000ns	0x9a98498	95.7223	89.8468
64000000000ns	0x9a985d8	93.7493	136.96
64000000000ns	0x9a98718	94.2503	115.345
64000000000ns	0x9a98858	103.993	121.786
64000000000ns	0x9a98998	61.6543	123.36
64000000000ns	0x9a98ad8	101.2	100.889
64000000000ns	0x9a98c18	79.7306	103.88
64000000000ns	0x9a98d58	104.099	104.315
64000000000ns	0x9a98e98	85.4541	109.807
64000000000ns	0x9a98fd8	101.033	85.8189
64000000000ns	0x9a99118	86.0841	70.2597
64000000000ns	0x9a99258	90.5126	114.017
64000000000ns	0x9a99398	86.4204	123.921
66000000000ns	0x9a96da8	121.856	75.6387
66000000000ns	0x9a96f98	101.569	99.3671
66000000000ns	0x9a97080	72.3425	98.5492
66000000000ns	0x9a97188	90.8597	101.331
66000000000ns	0x9a972b0	100.1	102.527
66000000000ns	0x9a973d8	123.402	79.0222
66000000000ns	0x9a97500	107.619	112.881
66000000000ns	0x9a97628	85.1471	92.7834
66000000000ns	0x9a97750	107.467	118.193
66000000000ns	0x9a97878	89.3268	90.1407
66000000000ns	0x9a979a0	100.2	84.6993
66000000000ns	0x9a97ac8	96.5693	95.5498
66000000000ns	0x9a97bf0	104.795	134.083
66000000000ns	0x9a97d18	106.86	91.2346
66000000000ns	0x9a97e58	125.939	91.2729
66000000000ns	0x9a97f98	109.288	101.514
66000000000ns	0x9a980d8	128.057	99.2294
66000000000ns	0x9a98218	124.602	76.0846
66000000000ns	0x9a98358	131.917	111.058
66000000000ns	0x9a98498	97.692	89.5001
66000000000ns	0x9a985d8	92.3983	138.435
66000000000ns	0x9a98718	94.079	117.337
66000000000ns	0x9a98858	102.921	123.473

66000000000ns	0x9a98998	59.7121	122.882
66000000000ns	0x9a98ad8	99.5139	99.8137
66000000000ns	0x9a98c18	81.3076	102.65
66000000000ns	0x9a98d58	105.767	105.418
66000000000ns	0x9a98e98	86.1276	107.924
66000000000ns	0x9a98fd8	99.1299	85.2049
66000000000ns	0x9a99118	86.0882	72.2597
66000000000ns	0x9a99258	89.4377	112.331
66000000000ns	0x9a99398	84.59	124.727
68000000000ns	0x9a96da8	121.34	77.5712
68000000000ns	0x9a96f98	100.81	97.5165
68000000000ns	0x9a97080	70.3632	98.2622
68000000000ns	0x9a97188	92.2552	99.8985
68000000000ns	0x9a972b0	98.1753	101.983
68000000000ns	0x9a973d8	124.604	80.62
68000000000ns	0x9a97500	109.239	114.055
68000000000ns	0x9a97628	84.9884	94.7771
68000000000ns	0x9a97750	109.458	117.999
68000000000ns	0x9a97878	87.5672	91.0914
68000000000ns	0x9a979a0	99.3954	82.8681
68000000000ns	0x9a97ac8	96.6468	97.5483
68000000000ns	0x9a97bf0	106.308	132.774
68000000000ns	0x9a97d18	107.098	93.2204
68000000000ns	0x9a97e58	124.031	90.6734
68000000000ns	0x9a97f98	107.475	102.359
68000000000ns	0x9a980d8	128.431	101.194
68000000000ns	0x9a98218	125.755	74.4506
68000000000ns	0x9a98358	130.771	112.697
68000000000ns	0x9a98498	99.3863	88.4373
68000000000ns	0x9a985d8	93.0424	136.541
68000000000ns	0x9a98718	93.1961	119.132
68000000000ns	0x9a98858	102.055	125.276
68000000000ns	0x9a98998	58.3537	121.414
68000000000ns	0x9a98ad8	101.306	98.9259
68000000000ns	0x9a98c18	82.9723	101.541
68000000000ns	0x9a98d58	106.566	107.252
68000000000ns	0x9a98e98	84.6926	106.53
68000000000ns	0x9a98fd8	97.3074	86.0287
68000000000ns	0x9a99118	86.195	70.2625
68000000000ns	0x9a99258	88.185	113.89
68000000000ns	0x9a99398	83.7477	126.541
70000000000ns	0x9a96da8	122.823	78.9131
70000000000ns	0x9a96f98	102.466	98.6387
70000000000ns	0x9a97080	71.6949	96.77
70000000000ns	0x9a97188	94.186	99.377

70000000000ns	0x9a972b0	96.8344	103.467
70000000000ns	0x9a973d8	126.542	80.1229
70000000000ns	0x9a97500	107.348	114.707
70000000000ns	0x9a97628	85.3504	96.7441
70000000000ns	0x9a97750	109.322	119.995
70000000000ns	0x9a97878	86.2307	92.5793
70000000000ns	0x9a979a0	99.6635	84.8501
70000000000ns	0x9a97ac8	94.7576	96.892
70000000000ns	0x9a97bf0	106.278	130.775
70000000000ns	0x9a97d18	108.993	92.5797
70000000000ns	0x9a97e58	122.291	91.659
70000000000ns	0x9a97f98	106.122	103.832
70000000000ns	0x9a980d8	129.541	99.5308
70000000000ns	0x9a98218	125.169	72.5383
70000000000ns	0x9a98358	128.795	112.39
70000000000ns	0x9a98498	100.687	86.9184
70000000000ns	0x9a985d8	91.7544	138.071
70000000000ns	0x9a98718	91.2629	118.619
70000000000ns	0x9a98858	103.975	124.715
70000000000ns	0x9a98998	59.9345	120.189
70000000000ns	0x9a98ad8	101.09	100.914
70000000000ns	0x9a98c18	83.2504	103.522
70000000000ns	0x9a98d58	108.53	107.628
70000000000ns	0x9a98e98	83.925	104.684
70000000000ns	0x9a98fd8	99.1688	86.7603
70000000000ns	0x9a99118	84.6518	68.9902
70000000000ns	0x9a99258	86.3602	113.071
70000000000ns	0x9a99398	82.1457	125.344
72000000000ns	0x9a96da8	124.337	77.6065
72000000000ns	0x9a96f98	104.291	97.8215
72000000000ns	0x9a97080	69.7016	96.9333
72000000000ns	0x9a97188	93.0395	97.7382
72000000000ns	0x9a972b0	98.6966	104.197
72000000000ns	0x9a973d8	126.548	78.1229
72000000000ns	0x9a97500	105.349	114.683
72000000000ns	0x9a97628	83.6167	95.7469
72000000000ns	0x9a97750	110.403	118.313
72000000000ns	0x9a97878	84.4427	91.6832
72000000000ns	0x9a979a0	100.164	86.7864
72000000000ns	0x9a97ac8	96.6855	96.3601
72000000000ns	0x9a97bf0	107.351	132.463
72000000000ns	0x9a97d18	107.418	93.8122
72000000000ns	0x9a97e58	120.424	92.3761
72000000000ns	0x9a97f98	104.16	104.217
72000000000ns	0x9a980d8	129.359	97.5391

72000000000ns	0x9a98218	123.983	70.9281
72000000000ns	0x9a98358	127.006	113.284
72000000000ns	0x9a98498	100.267	88.8738
72000000000ns	0x9a985d8	93.2134	139.439
72000000000ns	0x9a98718	92.4774	120.208
72000000000ns	0x9a98858	105.377	126.142
72000000000ns	0x9a98998	61.62	121.266
72000000000ns	0x9a98ad8	103.09	100.903
72000000000ns	0x9a98c18	82.6373	105.426
72000000000ns	0x9a98d58	108.409	109.624
72000000000ns	0x9a98e98	84.0439	102.687
72000000000ns	0x9a98fd8	97.5931	85.5286
72000000000ns	0x9a99118	83.9023	67.136
72000000000ns	0x9a99258	85.5602	114.904
72000000000ns	0x9a99398	81.9218	123.356
74000000000ns	0x9a96da8	125.593	79.1635
74000000000ns	0x9a96f98	102.393	98.4501
74000000000ns	0x9a97080	71.1416	95.5454
74000000000ns	0x9a97188	93.5916	99.6605
74000000000ns	0x9a972b0	96.8625	103.399
74000000000ns	0x9a973d8	125.985	76.2038
74000000000ns	0x9a97500	106.78	116.079
74000000000ns	0x9a97628	83.5861	93.7472
74000000000ns	0x9a97750	109.658	120.168
74000000000ns	0x9a97878	82.4476	91.544
74000000000ns	0x9a979a0	98.2148	86.3395
74000000000ns	0x9a97ac8	96.0645	94.459
74000000000ns	0x9a97bf0	107.923	134.379
74000000000ns	0x9a97d18	105.449	93.4625
74000000000ns	0x9a97e58	119.775	90.4843
74000000000ns	0x9a97f98	105.049	106.009
74000000000ns	0x9a980d8	128.394	95.7871
74000000000ns	0x9a98218	122.15	71.728
74000000000ns	0x9a98358	125.374	112.128
74000000000ns	0x9a98498	98.5352	87.8739
74000000000ns	0x9a985d8	94.932	140.462
74000000000ns	0x9a98718	92.3495	118.213
74000000000ns	0x9a98858	107.363	126.373
74000000000ns	0x9a98998	63.0677	119.886
74000000000ns	0x9a98ad8	104.66	99.6633
74000000000ns	0x9a98c18	80.6943	104.951
74000000000ns	0x9a98d58	110.403	109.774
74000000000ns	0x9a98e98	83.2427	104.52
74000000000ns	0x9a98fd8	95.961	86.6845
74000000000ns	0x9a99118	85.8883	66.9001

74000000000ns	0x9a99258	87.5548	114.758
74000000000ns	0x9a99398	80.0975	124.176
76000000000ns	0x9a96da8	124.018	80.3968
76000000000ns	0x9a96f98	104.111	99.4734
76000000000ns	0x9a97080	73.0418	96.1692
76000000000ns	0x9a97188	94.8324	101.229
76000000000ns	0x9a972b0	97.1542	105.378
76000000000ns	0x9a973d8	124.307	75.1148
76000000000ns	0x9a97500	108.07	114.551
76000000000ns	0x9a97628	82.9402	91.8543
76000000000ns	0x9a97750	108.404	118.61
76000000000ns	0x9a97878	81.5077	93.3094
76000000000ns	0x9a979a0	99.2942	88.0232
76000000000ns	0x9a97ac8	94.1779	95.1231
76000000000ns	0x9a97bf0	105.934	134.162
76000000000ns	0x9a97d18	106.709	95.0151
76000000000ns	0x9a97e58	120.827	92.1849
76000000000ns	0x9a97f98	103.582	104.649
76000000000ns	0x9a980d8	128.368	93.7873
76000000000ns	0x9a98218	123.057	69.9455
76000000000ns	0x9a98358	126.909	110.846
76000000000ns	0x9a98498	100.221	88.9494
76000000000ns	0x9a985d8	94.6907	142.447
76000000000ns	0x9a98718	92.0462	120.189
76000000000ns	0x9a98858	105.827	127.653
76000000000ns	0x9a98998	61.1783	120.542
76000000000ns	0x9a98ad8	102.671	99.8753
76000000000ns	0x9a98c18	81.695	106.683
76000000000ns	0x9a98d58	110.57	111.767
76000000000ns	0x9a98e98	85.0783	105.314
76000000000ns	0x9a98fd8	97.3467	85.2424
76000000000ns	0x9a99118	87.7246	66.1076
76000000000ns	0x9a99258	89.0998	113.488
76000000000ns	0x9a99398	78.1239	124.5
78000000000ns	0x9a96da8	125.917	79.767
78000000000ns	0x9a96f98	104.707	101.382
78000000000ns	0x9a97080	73.6946	94.2787
78000000000ns	0x9a97188	95.5201	103.107
78000000000ns	0x9a972b0	98.405	106.939
78000000000ns	0x9a973d8	125.215	76.8968
78000000000ns	0x9a97500	109.971	115.171
78000000000ns	0x9a97628	84.305	93.3163
78000000000ns	0x9a97750	109.183	116.768
78000000000ns	0x9a97878	81.5749	95.3083
78000000000ns	0x9a979a0	97.9832	89.5336

78000000000ns	0x9a97ac8	92.2212	95.537
78000000000ns	0x9a97bf0	104.842	135.838
78000000000ns	0x9a97d18	106.713	97.0151
78000000000ns	0x9a97e58	120.733	94.1826
78000000000ns	0x9a97f98	103.777	106.64
78000000000ns	0x9a980d8	126.37	93.869
78000000000ns	0x9a98218	124.81	70.9069
78000000000ns	0x9a98358	128.878	111.196
78000000000ns	0x9a98498	101.233	90.6744
78000000000ns	0x9a985d8	93.707	144.189
78000000000ns	0x9a98718	93.4104	121.652
78000000000ns	0x9a98858	107.732	127.045
78000000000ns	0x9a98998	59.6132	121.787
78000000000ns	0x9a98ad8	102.685	97.8754
78000000000ns	0x9a98c18	79.8032	106.034
78000000000ns	0x9a98d58	111.117	109.843
78000000000ns	0x9a98e98	84.9548	107.31
78000000000ns	0x9a98fd8	95.792	83.9843
78000000000ns	0x9a99118	86.9537	67.953
78000000000ns	0x9a99258	87.1644	112.983
78000000000ns	0x9a99398	79.9719	123.735
80000000000ns	0x9a96da8	126.837	77.9914
80000000000ns	0x9a96f98	106.602	100.742
80000000000ns	0x9a97080	73.0781	92.3761
80000000000ns	0x9a97188	94.1763	104.588
80000000000ns	0x9a972b0	99.0685	108.825
80000000000ns	0x9a973d8	123.219	76.7719
80000000000ns	0x9a97500	111.596	114.004
80000000000ns	0x9a97628	86.2423	93.813
80000000000ns	0x9a97750	108.946	118.754
80000000000ns	0x9a97878	83.3737	96.1824
80000000000ns	0x9a979a0	99.9707	89.3108
80000000000ns	0x9a97ac8	93.1844	97.2898
80000000000ns	0x9a97bf0	103.5	134.355
80000000000ns	0x9a97d18	106.012	95.1419
80000000000ns	0x9a97e58	122.732	94.1126
80000000000ns	0x9a97f98	102.978	104.806
80000000000ns	0x9a980d8	124.799	95.1076
80000000000ns	0x9a98218	123.575	69.3338
80000000000ns	0x9a98358	130.125	112.76
80000000000ns	0x9a98498	101.492	88.6912
80000000000ns	0x9a985d8	95.6938	143.959
80000000000ns	0x9a98718	91.421	121.857
80000000000ns	0x9a98858	108.906	128.664
80000000000ns	0x9a98998	58.2816	120.295

80000000000ns	0x9a98ad8	102.935	95.8911
80000000000ns	0x9a98c18	78.677	107.687
80000000000ns	0x9a98d58	113.032	110.421
80000000000ns	0x9a98e98	83.4325	108.607
80000000000ns	0x9a98fd8	94.6906	82.3149
80000000000ns	0x9a99118	87.2911	65.9817
80000000000ns	0x9a99258	85.2534	113.574
80000000000ns	0x9a99398	78.0188	123.304
82000000000ns	0x9a96da8	124.857	78.2762
82000000000ns	0x9a96f98	107.929	99.2458
82000000000ns	0x9a97080	73.7405	90.489
82000000000ns	0x9a97188	96.0407	105.312
82000000000ns	0x9a972b0	97.8416	107.246
82000000000ns	0x9a973d8	121.355	77.4957
82000000000ns	0x9a97500	113.477	113.324
82000000000ns	0x9a97628	84.2439	93.734
82000000000ns	0x9a97750	110.852	118.15
82000000000ns	0x9a97878	81.3746	96.1255
82000000000ns	0x9a979a0	98.0592	89.899
82000000000ns	0x9a97ac8	91.8886	95.7663
82000000000ns	0x9a97bf0	103.409	136.353
82000000000ns	0x9a97d18	104.578	93.7481
82000000000ns	0x9a97e58	122.075	96.0016
82000000000ns	0x9a97f98	102.135	106.62
82000000000ns	0x9a980d8	126.501	96.1579
82000000000ns	0x9a98218	124.408	71.1521
82000000000ns	0x9a98358	128.144	113.039
82000000000ns	0x9a98498	103.232	89.6769
82000000000ns	0x9a985d8	97.5204	143.145
82000000000ns	0x9a98718	91.8172	119.896
82000000000ns	0x9a98858	110.906	128.63
82000000000ns	0x9a98998	56.2866	120.153
82000000000ns	0x9a98ad8	100.954	96.1713
82000000000ns	0x9a98c18	80.0864	109.106
82000000000ns	0x9a98d58	115.026	110.264
82000000000ns	0x9a98e98	81.6056	109.421
82000000000ns	0x9a98fd8	93.6655	84.0321
82000000000ns	0x9a99118	87.1857	67.9789
82000000000ns	0x9a99258	84.4587	111.738
82000000000ns	0x9a99398	76.0388	123.022
84000000000ns	0x9a96da8	123.446	76.8597
84000000000ns	0x9a96f98	107.446	97.3051
84000000000ns	0x9a97080	74.6237	88.6946
84000000000ns	0x9a97188	94.167	106.012
84000000000ns	0x9a972b0	97.3032	105.32

84000000000ns	0x9a973d8	121.612	79.4791
84000000000ns	0x9a97500	112.209	114.871
84000000000ns	0x9a97628	86.0761	92.9323
84000000000ns	0x9a97750	108.86	117.979
84000000000ns	0x9a97878	83.0796	95.0801
84000000000ns	0x9a979a0	96.7932	91.4474
84000000000ns	0x9a97ac8	92.8992	94.0404
84000000000ns	0x9a97bf0	104.164	134.501
84000000000ns	0x9a97d18	102.583	93.6084
84000000000ns	0x9a97e58	120.118	96.417
84000000000ns	0x9a97f98	104.031	105.985
84000000000ns	0x9a980d8	127.45	97.9188
84000000000ns	0x9a98218	125.429	69.4321
84000000000ns	0x9a98358	129.881	112.048
84000000000ns	0x9a98498	103.057	91.6691
84000000000ns	0x9a985d8	99.3687	143.909
84000000000ns	0x9a98718	89.8172	119.897
84000000000ns	0x9a98858	112.016	126.966
84000000000ns	0x9a98998	54.3566	119.629
84000000000ns	0x9a98ad8	98.9588	96.3056
84000000000ns	0x9a98c18	81.5351	107.727
84000000000ns	0x9a98d58	116.427	111.691
84000000000ns	0x9a98e98	80.5128	111.096
84000000000ns	0x9a98fd8	92.948	85.899
84000000000ns	0x9a99118	86.812	66.0141
84000000000ns	0x9a99258	82.6509	110.883
84000000000ns	0x9a99398	77.5426	121.703
86000000000ns	0x9a96da8	124.408	78.613
86000000000ns	0x9a96f98	105.966	98.6504
86000000000ns	0x9a97080	74.1615	86.7487
86000000000ns	0x9a97188	92.5629	107.206
86000000000ns	0x9a972b0	95.9788	103.821
86000000000ns	0x9a973d8	123.501	80.1345
86000000000ns	0x9a97500	114.136	115.407
86000000000ns	0x9a97628	87.1874	94.5951
86000000000ns	0x9a97750	110.225	116.517
86000000000ns	0x9a97878	84.8367	94.1249
86000000000ns	0x9a979a0	97.2521	89.5007
86000000000ns	0x9a97ac8	90.9405	94.4446
86000000000ns	0x9a97bf0	105.972	135.355
86000000000ns	0x9a97d18	100.758	94.4272
86000000000ns	0x9a97e58	118.119	96.4633
86000000000ns	0x9a97f98	105.485	107.359
86000000000ns	0x9a980d8	126.176	96.3768
86000000000ns	0x9a98218	123.684	70.4105

86000000000ns	0x9a98358	128.161	113.068
86000000000ns	0x9a98498	104.481	90.2649
86000000000ns	0x9a985d8	99.1326	145.895
86000000000ns	0x9a98718	89.1036	118.029
86000000000ns	0x9a98858	112.642	128.866
86000000000ns	0x9a98998	56.0285	120.726
86000000000ns	0x9a98ad8	98.2747	98.1849
86000000000ns	0x9a98c18	79.6633	108.432
86000000000ns	0x9a98d58	114.848	110.463
86000000000ns	0x9a98e98	78.6162	111.731
86000000000ns	0x9a98fd8	91.9768	87.6474
86000000000ns	0x9a99118	88.8119	65.9923
86000000000ns	0x9a99258	84.2794	112.044
86000000000ns	0x9a99398	77.3145	119.716
88000000000ns	0x9a96da8	122.776	79.7692
88000000000ns	0x9a96f98	106.81	96.8374
88000000000ns	0x9a97080	76.0859	87.2934
88000000000ns	0x9a97188	94.0546	105.874
88000000000ns	0x9a972b0	96.6478	101.936
88000000000ns	0x9a973d8	124.802	81.6539
88000000000ns	0x9a97500	116.023	114.745
88000000000ns	0x9a97628	86.149	92.8858
88000000000ns	0x9a97750	109.477	114.663
88000000000ns	0x9a97878	83.4524	95.5684
88000000000ns	0x9a979a0	99.222	89.1552
88000000000ns	0x9a97ac8	89.2017	95.433
88000000000ns	0x9a97bf0	107.551	134.127
88000000000ns	0x9a97d18	99.1989	93.1749
88000000000ns	0x9a97e58	119.88	95.5157
88000000000ns	0x9a97f98	104.592	105.569
88000000000ns	0x9a980d8	124.847	97.8709
88000000000ns	0x9a98218	122.756	72.182
88000000000ns	0x9a98358	126.599	114.317
88000000000ns	0x9a98498	102.615	89.5455
88000000000ns	0x9a985d8	101.125	145.724
88000000000ns	0x9a98718	88.944	116.035
88000000000ns	0x9a98858	114.269	127.703
88000000000ns	0x9a98998	58.019	120.532
88000000000ns	0x9a98ad8	97.3044	99.9338
88000000000ns	0x9a98c18	78.8457	110.257
88000000000ns	0x9a98d58	115.563	112.331
88000000000ns	0x9a98e98	80.5783	112.118
88000000000ns	0x9a98fd8	93.9304	87.2188
88000000000ns	0x9a99118	90.7742	66.3787
88000000000ns	0x9a99258	85.3149	113.755

88000000000ns	0x9a99398	79.3031	119.503
90000000000ns	0x9a96da8	124.494	80.7919
90000000000ns	0x9a96f98	107.603	98.6736
90000000000ns	0x9a97080	78.0822	87.4151
90000000000ns	0x9a97188	95.5215	107.233
90000000000ns	0x9a972b0	94.7908	101.194
90000000000ns	0x9a973d8	123.607	83.2582
90000000000ns	0x9a97500	118.002	115.033
90000000000ns	0x9a97628	84.2561	93.5313
90000000000ns	0x9a97750	107.515	114.276
90000000000ns	0x9a97878	82.4674	97.309
90000000000ns	0x9a979a0	98.2685	90.9133
90000000000ns	0x9a97ac8	89.1951	97.433
90000000000ns	0x9a97bf0	108.52	135.876
90000000000ns	0x9a97d18	100.256	94.8724
90000000000ns	0x9a97e58	118.012	96.2286
90000000000ns	0x9a97f98	104.003	103.658
90000000000ns	0x9a980d8	125.471	99.7709
90000000000ns	0x9a98218	122.681	74.1806
90000000000ns	0x9a98358	126.945	116.287
90000000000ns	0x9a98498	104.598	89.8013
90000000000ns	0x9a985d8	103.067	146.204
90000000000ns	0x9a98718	89.3808	117.987
90000000000ns	0x9a98858	115.563	129.228
90000000000ns	0x9a98998	59.1197	122.202
90000000000ns	0x9a98ad8	98.8753	101.172
90000000000ns	0x9a98c18	77.7537	111.932
90000000000ns	0x9a98d58	116.905	113.814
90000000000ns	0x9a98e98	78.6116	111.755
90000000000ns	0x9a98fd8	92.3032	88.3817
90000000000ns	0x9a99118	91.5669	68.2149
90000000000ns	0x9a99258	85.8961	115.668
90000000000ns	0x9a99398	79.1694	117.508
92000000000ns	0x9a96da8	123.547	82.5532
92000000000ns	0x9a96f98	109.255	99.8006
92000000000ns	0x9a97080	76.8866	85.8118
92000000000ns	0x9a97188	96.2265	105.362
92000000000ns	0x9a972b0	92.7968	101.039
92000000000ns	0x9a973d8	123.259	81.2888
92000000000ns	0x9a97500	116.217	115.935
92000000000ns	0x9a97628	82.3924	94.2571
92000000000ns	0x9a97750	107.912	116.236
92000000000ns	0x9a97878	84.4659	97.3878
92000000000ns	0x9a979a0	97.7388	92.8419
92000000000ns	0x9a97ac8	88.1332	99.1279

92000000000ns	0x9a97bf0	107.59	137.647
92000000000ns	0x9a97d18	98.3251	94.353
92000000000ns	0x9a97e58	116.765	94.6645
92000000000ns	0x9a97f98	102.729	102.116
92000000000ns	0x9a980d8	123.649	100.595
92000000000ns	0x9a98218	120.695	73.9433
92000000000ns	0x9a98358	127.908	114.533
92000000000ns	0x9a98498	104.935	91.7727
92000000000ns	0x9a985d8	101.648	147.613
92000000000ns	0x9a98718	90.7055	119.485
92000000000ns	0x9a98858	117.552	129.435
92000000000ns	0x9a98998	58.3777	124.059
92000000000ns	0x9a98ad8	97.9969	102.968
92000000000ns	0x9a98c18	79.3118	113.186
92000000000ns	0x9a98d58	115.469	115.206
92000000000ns	0x9a98e98	78.8651	109.771
92000000000ns	0x9a98fd8	91.2654	86.672
92000000000ns	0x9a99118	93.3656	69.0893
92000000000ns	0x9a99258	87.7914	116.307
92000000000ns	0x9a99398	78.2264	119.272
94000000000ns	0x9a96da8	123.968	80.598
94000000000ns	0x9a96f98	110.986	98.7983
94000000000ns	0x9a97080	74.9784	86.4105
94000000000ns	0x9a97188	97.4202	106.966
94000000000ns	0x9a972b0	92.4033	99.0782
94000000000ns	0x9a973d8	122.999	83.2719
94000000000ns	0x9a97500	118.05	115.135
94000000000ns	0x9a97628	84.0871	95.3192
94000000000ns	0x9a97750	109.077	117.861
94000000000ns	0x9a97878	82.6763	96.4948
94000000000ns	0x9a979a0	99.7387	92.8167
94000000000ns	0x9a97ac8	86.1593	99.4498
94000000000ns	0x9a97bf0	108.669	135.963
94000000000ns	0x9a97d18	99.9821	95.4729
94000000000ns	0x9a97e58	115.211	93.406
94000000000ns	0x9a97f98	101.02	103.154
94000000000ns	0x9a980d8	121.936	99.5639
94000000000ns	0x9a98218	118.936	74.8953
94000000000ns	0x9a98358	129.788	115.215
94000000000ns	0x9a98498	105.859	93.5465
94000000000ns	0x9a985d8	99.6878	147.214
94000000000ns	0x9a98718	92.4147	120.524
94000000000ns	0x9a98858	116.924	127.536
94000000000ns	0x9a98998	56.7882	122.845
94000000000ns	0x9a98ad8	99.9383	102.488

94000000000ns	0x9a98c18	81.1074	112.305
94000000000ns	0x9a98d58	113.857	116.39
94000000000ns	0x9a98e98	77.2735	110.982
94000000000ns	0x9a98fd8	89.4596	87.5317
94000000000ns	0x9a99118	95.1661	69.9601
94000000000ns	0x9a99258	86.6721	117.965
94000000000ns	0x9a99398	78.5859	121.239
96000000000ns	0x9a96da8	122.707	79.0458
96000000000ns	0x9a96f98	109.548	97.4077
96000000000ns	0x9a97080	76.9693	86.6003
96000000000ns	0x9a97188	97.8379	108.922
96000000000ns	0x9a972b0	90.4609	99.5548
96000000000ns	0x9a973d8	123.209	85.2608
96000000000ns	0x9a97500	116.543	113.82
96000000000ns	0x9a97628	82.908	93.7038
96000000000ns	0x9a97750	110.055	116.117
96000000000ns	0x9a97878	83.5352	94.6886
96000000000ns	0x9a979a0	101.676	93.3139
96000000000ns	0x9a97ac8	87.0733	101.229
96000000000ns	0x9a97bf0	109.672	137.693
96000000000ns	0x9a97d18	101.982	95.434
96000000000ns	0x9a97e58	117.158	92.948
96000000000ns	0x9a97f98	102.717	104.212
96000000000ns	0x9a980d8	120.015	100.121
96000000000ns	0x9a98218	117.751	73.2843
96000000000ns	0x9a98358	127.827	114.823
96000000000ns	0x9a98498	107.859	93.5578
96000000000ns	0x9a985d8	101.673	146.968
96000000000ns	0x9a98718	91.0559	119.056
96000000000ns	0x9a98858	118.224	129.056
96000000000ns	0x9a98998	57.9042	121.185
96000000000ns	0x9a98ad8	101.934	102.615
96000000000ns	0x9a98c18	79.3547	111.342
96000000000ns	0x9a98d58	113.694	114.397
96000000000ns	0x9a98e98	75.2746	110.915
96000000000ns	0x9a98fd8	88.366	89.2062
96000000000ns	0x9a99118	96.1622	68.2258
96000000000ns	0x9a99258	84.6989	118.291
96000000000ns	0x9a99398	76.7089	120.549
98000000000ns	0x9a96da8	124.58	79.7454
98000000000ns	0x9a96f98	111.421	98.1103
98000000000ns	0x9a97080	78.1708	88.1992
98000000000ns	0x9a97188	95.8653	108.593
98000000000ns	0x9a972b0	92.3069	100.324
98000000000ns	0x9a973d8	123.401	87.2516

98000000000ns	0x9a97500	118.541	113.912
98000000000ns	0x9a97628	84.5431	94.8555
98000000000ns	0x9a97750	110.891	114.3
98000000000ns	0x9a97878	85.1969	95.8016
98000000000ns	0x9a979a0	99.9872	94.3856
98000000000ns	0x9a97ac8	89.0022	101.757
98000000000ns	0x9a97bf0	111.66	137.909
98000000000ns	0x9a97d18	102.528	97.358
98000000000ns	0x9a97e58	117.155	90.948
98000000000ns	0x9a97f98	102.432	102.232
98000000000ns	0x9a980d8	121.387	98.6663
98000000000ns	0x9a98218	116.526	74.8652
98000000000ns	0x9a98358	125.853	115.143
98000000000ns	0x9a98498	106.406	92.1831
98000000000ns	0x9a985d8	100.232	148.355
98000000000ns	0x9a98718	89.1601	119.693
98000000000ns	0x9a98858	120.07	129.826
98000000000ns	0x9a98998	58.1231	119.197
98000000000ns	0x9a98ad8	102.97	100.904
98000000000ns	0x9a98c18	80.9129	110.088
98000000000ns	0x9a98d58	115.636	113.922
98000000000ns	0x9a98e98	73.4779	110.037
98000000000ns	0x9a98fd8	90.2915	88.6655
98000000000ns	0x9a99118	96.6465	66.2854
98000000000ns	0x9a99258	86.6988	118.305
98000000000ns	0x9a99398	74.8021	119.945